ITHEA

International Journal INFORMATION THEORIES & APPLICATIONS

2003 Volume 10 Number 4

INFORMATION THEORIES & APPLICATIONS

ISSN 1310-0513
Volume 10 / 2003, Number 4

IJ ITA is official publisher of the scientific papers of the members of the Association of Developers and Users of Intellectualized Systems (ADUIS).

IJ ITA welcomes scientific papers connected with any information theory or its application.
Original and non-standard ideas will be published with preferences.

Papers must be written in English.

Responsibility for papers published in IJ ITA belongs to authors.
Please get permission to reprint any copyrighted material before you send it to IJ ITA.

IJ ITA rules for preparing the manuscripts are compulsory.
The rules for the papers for IJ ITA as well as the subscription fees are given on **www.foibg.com/ijita** .

The camera-ready copy of the paper should be received by e-mail: foi@nlcv.net

International Journal "INFORMATION THEORIES & APPLICATIONS"
Vol.10, Number 4, 2003
Printed in Bulgaria
Edited by the Institute of Information Theories and Applications FOI ITHEA, Bulgaria
Publisher: FOI-COMMERCE - Sofia, 1000, P.O.B. 775, Bulgaria
www.foibg.com e-mail: foi@nlcv.net

DIFFERENTIAL BALANCED TREES AND (0,1) MATRICES[1]

H. Sahakyan, L. Aslanyan

Abstract: Links and similarities between the combinatorial optimization problems and the hierarchical search algorithms are discussed. One is the combinatorial greedy algorithm of step-by-step construction of the column-constraint (0,1) matrices with the different rows. The second is the base search construction of databases, - the class of the well known weight-balanced binary trees. Noted, that in some approximation each of the above problems might be interpreted in terms of the second problem. The constraints in matrices imply the novel concept of a differential balance in hierarchical trees. The obtained results extend the knowledge for balanced trees and prove that the known greedy algorithm for matrices is applicable in the world of balanced trees providing optimization on trees in layers.

Keywords: search, balanced trees, (0,1)-matrices, greedy algorithm.

1. Introduction

In this paper a new class of weight-balanced trees [K, 1973] is introduced and investigated. In some sense these are extensions of the concept of the bounded balanced trees. Bounded balanced trees were analysed in various publications, e.g. [R, 1977], being the main data structure of search in dynamic databases. In Section 2 below the height estimate for bounded-balanced trees is considered and an estimate for the weight-balanced trees with the newly introduced differential balances and constraints is obtained.

The theory of weight-balanced trees is very rich. Practically this is also the base model of hierarchical search and decision support. In search, several restrictions in terms of balances are applied in a dynamic environment with insertion of new and deletion of obsolete search elements. The balances in nodes are under the change during this process. In a dis-balanced node rotations are used to correct the situation. Several queries, related to these models are traditional. Which is the tree height in a given balance and in a given set of search elements? Which is the average path length in a search tree? A particular new postulation is the following. Is it possible to construct a tree or to construct all the trees that may appear in a search model with the given constraints? This is a particular interest of the current paper.

The stated problem will be studied in several extensions, which are also a typical element of search models. E.g. - some specific classes of balanced trees, called trees of bounded heights, introduced in [A, 1989], [A, 1999].

The concept of bounded-balance is extended in Section 3, defining layer-constraint balanced trees. The idea of layer-constraints is then developed in Section 4, considering a practical extension of the concept of weight-balanced trees - defining summary balances for tree layers. This structure is related to mentioned combinatorial problem – constructing the constraint based (0,1)-matrices with different rows. In [S, 1986], [S, 1995] a greedy algorithm is constructed for solving the mentioned combinatorial problem and it is proven optimal in local steps. The algorithm for solving this problem is reducible to the constructing of weight-balanced trees by the given summary differential balances in layers. Similarly, in the world of balanced trees this proves a heuristic optimization on trees in layers.

2. Bounded-balanced Trees

Let T_m be a non-empty extended binary tree [R, 1977] with m leaves, and T_l and T_r are the left and right root-subtrees of T_m. We denote by l and r the numbers of leaves of T_l and T_r (called weights) and assume that $l > 0$ and $r > 0$. Then $m = l + r$.

Definition [R, 1977]. The fraction l / m is called the balance (left, fractional) of T_m in root vertex, being denoted by $\beta(T_m)$. $\beta(T_m)$ expresses the ratio weight of the left root-subtree and it obeys the condition $0 < \beta(T_m) < 1$.

[1] The research was supported by INTAS 00-397 and 00-626 Projects.

Definition [R, 1977]. For a given α, $0 \le \alpha \le 1/2$, T_m is called an α-balanced tree (or a tree from $WB[\alpha]$) if

1) $\alpha \le \beta(T_m) \le 1-\alpha$,
2) left and right subtrees of T_m belong to $WB[\alpha]$.

We assume by definition that the empty binary tree belongs to $WB[\alpha]$.

$WB[0]$ is the set of all binary trees and $WB[1/2]$ is the set of all perfectly balanced (with the equal left and right subtrees weights in each node) binary trees, and this is possible, when the number of leaves has the form 2^k $(k \ge 1)$.

The maximum possible height $h_\alpha(m)$ of trees from $WB[\alpha]$ is estimated in [R, 1977] – by the consideration of the most asymmetric trees of $WB[\alpha]$:

$$h_\alpha(m) \le \frac{\log m}{\log(1/(1-\alpha))}$$

(1)

It is also important to treat the question: given the binary trees with m leaves and with heights, restricted by a given number n, then - how “unbalanced” may be the trees, - which is the allowable minimum value for α? The answer (in a form of a sufficiency condition) is given by the lemma below, using the monotony of (1).

Lemma 1. If $\alpha \ge 1-\frac{1}{\sqrt[n]{m}}$, then $h_\alpha(m) \le n$.

The $\alpha \ge 1-\frac{1}{\sqrt[n]{m}}$ implies $\frac{1}{1-\alpha} \ge 2^{\frac{\log m}{n}}$, and then $\log(1/(1-\alpha)) \ge \frac{\log m}{n}$, and $\frac{\log m}{\log(1/(1-\alpha))} \le n$. For those α, $h_\alpha(m) \le \frac{\log m}{\log(1/(1-\alpha))} \le n$ by (1).

Now let us turn to the concept of balances in terms of differences of weights between subtrees.

Definition. The difference $r-l$ is called the differential balance (right) of T_m in the root vertex, denoted by $\delta(T_m)$. It obeys the following condition: $1-m < \delta(T_m) < m-1$.

Definition. For a given d, $0 \le d < m-1$, T_m is called differential-balanced tree with balance d (or a tree from $WDB[d]$) if

1) $-d \le \delta(T_m) \le d$,
2) left and right subtrees of T_m belong to $WDB[d]$.

The two balance schemes are tightly related. Let us formulate the base relations between the fractional and differential balances.

Let T_m be an extended binary tree with m leaves, and v_i is a vertex (not leaf) of T_m. We denote by m_i the weight of subtree rooted at v_i, and by l_i and r_i - the weights of its left and right subtrees, correspondingly. Starting at this point we will assume also, that l_i-s are not greater than r_i-s.

If T_m is an α-balanced tree then for each vertex v_i we have $\frac{l_i}{m_i} \geq \alpha$ by the definition. Let's estimate the weight differences between right and left subtrees for each v_i:

$r_i - l_i = m_i - 2l_i = m_i(1 - 2\frac{l_i}{m_i}) \leq m_i(1 - 2\alpha)$. Hence $r_i - l_i \leq \max_i m_i(1 - 2\alpha) = m(1 - 2\alpha)$.

Conclusion is that an α-balanced tree T_m is a differential balanced tree with $d = m(1 - 2\alpha)$.

Now let T_m is a differential balanced tree with balance d. For each vertex v_i, $r_i - l_i \leq d$ by the definition. Let's estimate the fraction $\frac{l_i}{m_i}$;

$$\frac{l_i}{m_i} = \frac{m_i - (r_i - l_i)}{2m_i} = \frac{1}{2}(1 - \frac{r_i - l_i}{m_i}) \geq \frac{1}{2}(1 - \frac{d}{m_i}) \geq \frac{1}{2}(1 - \frac{d}{m}).$$

Thus, a differential balanced tree T_m with balance d, is an α-balanced tree, with $\alpha = \frac{1}{2}(1 - \frac{d}{m})$. A more correct estimate is:

$$\alpha = \frac{1}{2}(1 - \max_i \frac{\min\{d, m_i - 2\}}{m_i}).$$

Next we consider the estimate of height of trees from $WDB[d]$, constructing the most asymmetric trees in this class. On each layer the subtrees with greatest weights have been partitioned into the subtrees with maximization of weights differences. Then the height estimate is the length of these “maximum weighted” branches.

On the first layer we get subtrees of weights $\frac{m+d}{2}$ and $\frac{m-d}{2}$. We will follow only the branch of weight $\frac{m+d}{2}$. On the next layer we will get a subtree of weight $\frac{\frac{m+d}{2} + d}{2} = \frac{m+3d}{4}$. In continuation, let k is the minimal index, where the maximal subtree weight becomes less than d. At that point the maximum weight doesn't exceed $\frac{m + (2^k - 1)d}{2^k}$.

$$\frac{m + (2^k - 1)d}{2^k} = \frac{m-d}{2^k} + d \text{, therefore } \frac{m-d}{2^k} < 1 \text{, and } k > \log(m - d).$$

Resuming, we receive, that after at most $\log(m - d) + 1$ steps the weight of maximal subtree is less than d. If $d \leq 1$, then the tree construction is complete, and we get a tree with the height estimate $\log(m - d) + 1$. Otherwise we continue the process, with the arbitrary partition of subtrees. At most $d - 1$ steps will be required. We receive the following final estimation – the heights of trees from $WDB[d]$ are restricted by $\log(m - d) + d$.

Now we treat the question about the constraints on balances when given that the heights are restricted. Let us consider the binary trees with m leaves, and heights, restricted by the given number n. The counterpart of Lemma 1 is the following proposition:

Lemma 2. If the differential balance d obeys: $\log(m-d)+d \le n$, then the height of tree with m leaves is restricted by n.

A practical note. The concept of differential balancing is reasonable to apply on trees as far as the weights of subtrees are greater than d, therefore - on layers of at most $\log(m-d)+1$ far from the root.

3. Layer-constrained Weight-balanced Trees

At this point the concept of differential balances is introduced and the general comparison with the base scheme – the weight-balanced trees is outlined. The particular properties of differential balances are that these are flexible on tree layers. The balance constraints may vary from layer to layer and/or the constraints might be given in terms of summary balances. In some cases it is important to apply these structures in the traditional case of the weight-balanced trees. These issues are considered below.

Definition. For a given α_i, $0 \le \alpha_i \le 1/2$, we say that T_m is α_i-balanced on layer i, if for each subtree T_{i_j} - rooted at layer i, $\alpha_i \le \beta(T_{i_j}) \le 1-\alpha_i$.

Definition. Given numbers $\alpha_0,\cdots,\alpha_k$, where $0 \le \alpha_i \le 1/2$, $i=0,\cdots,k$. We say that T_m is a tree from class $WB[\alpha_0,\cdots,\alpha_k]$, if T_m is α_i-balanced on layer i.

The leaves may be composite in $WB[\alpha_0,\cdots,\alpha_k]$ (when k-sequences are not enough to differentiate the nodes, the composite nodes may remain consisting of sets of virtual leaves). On the other hand, part of the balance values (a last portion) may be redundant. Consideration of the most asymmetric trees and paths in $WB[\alpha_0,\cdots,\alpha_k]$ gives the following estimation: the weights of subtrees (virtual at this point) of k-th layer are restricted in size by $(1-\alpha_0)(1-\alpha_1)\cdots(1-\alpha_k)m$. If there exists h, $h \le k$, such that $(1-\alpha_0)(1-\alpha_1)\cdots(1-\alpha_h)m \le 2$, then the height of the tree is restricted by h.

Now we consider layer constrained weight-balanced trees in sense of differential balances.

Definition. For a given d_i, $0 \le d_i < m-1$, we say that T_m has d_i differential balance on layer i, if for each subtree T_{i_j} - rooted at layer i, $-d_i \le \delta(T_{i_j}) \le d_i$.

Definition. Given numbers $d_0,\cdots,d_k$, where $0 \le d_i < m-1$, $i=0,\cdots,k$. We say that T_m is a tree from class $WDB[d_0,\cdots,d_k]$, if T_m has differential balances d_i on layers i.

Similarly with the class $WB[\alpha_0,\cdots,\alpha_k]$, the leaves may be composite, or some last balance values may be redundant for trees of $WDB[d_0,\cdots,d_k]$. Consider the most asymmetric trees of the class $WDB[d_0,\cdots,d_k]$. Using reasoning, similar to the used above, we get that the weights of subtrees on the k-th layer of trees are restricted by $\dfrac{m+d_0+2d_1+\cdots+2^k d_k}{2^{k+1}}$.

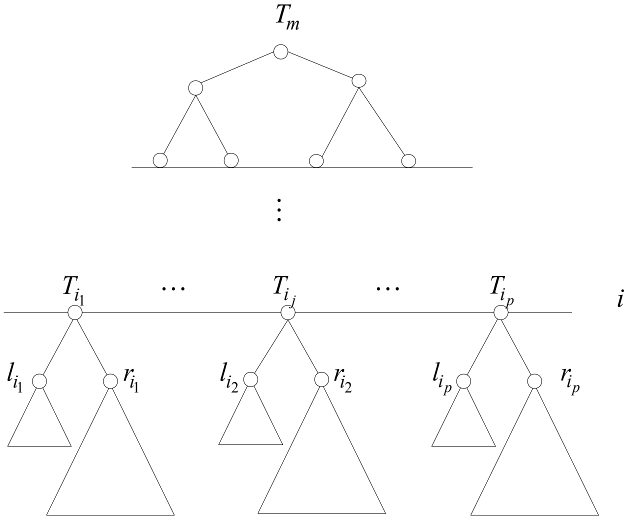
If there exists h, $h \le k$, such that $\frac{m + d_0 + 2d_1 + \cdots + 2^h d_h}{2^{h+1}} \le 1$, then the overall height of tree is restricted by h. It is easy to see that $d_0, \cdots, d_k$ must obey in this case very specific restrictions, which limits the selection and the meaning of differential balances.

4. Summary Differential Balanced Tress and (0,1)-matrices

Definition. Given numbers $D_0, \cdots, D_n$ where $0 \le D_i < m-1,\ i = 0, \cdots, n$. We say that T_m has D_i summary differential balance on i -th layer, if $R_i - L_i \le D_i$, where R_i is the sum of weights of the all right subtrees rooted at the layer i and L_i is the same sum for the left subtrees.

Definition. T_m is called $\{D_0, \cdots, D_n\}$ summary differential balanced tree if the summary balance on i -th layer equals to D_i, $i = 0, \cdots, n$.

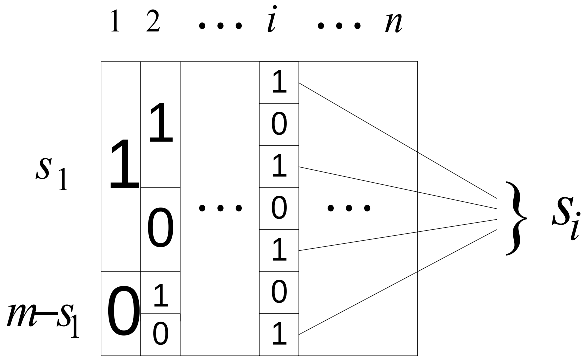
Let $T_{i_1}, \cdots, T_{i_p}$ are subtrees rooted at i -th layer having the weights $m_{i_1}, \cdots, m_{i_p}$ correspondingly, and let l_{i_j} and r_{i_j} are the weights of left and right subtrees of T_{i_j}. Then $R_i = \sum_{j=1}^{p} r_{i_j}$ and $L_i = \sum_{j=1}^{p} l_{i_j}$.



Usually the balance criterion restricts the weights of subtrees, making it possible to optimize the height of the tree. The summary balance allows any weights for subtrees, requiring only satisfying the given summary constraints on layers. This is a week form constraints for constructing an optimized decision tree. The most asymmetric trees are very diverse in this case. In next point the combinatorial origin of these differential schemes will be described. In classes of summary balanced trees the problem is in existence of a binary tree with the given characteristics $D_0, \cdots, D_n$, and in case of existence – the algorithmic construction of such trees. In special cases the issue of construction of trees might be the interest, when an additional functional for optimization is given. In particular, optimality might be required as for subtree weights on layers, for number of subtrees on layers, a special functional optimization, etc.

Here is the combinatorial counterpart of the scheme of the summary differential balances. Given an integer vector $S = (s_1, \cdots, s_n)$, where $0 \le s_i \le m$, $i = 1, \cdots, n$. The interest is in (0,1) -matrices of size $m \times n$ (m is the number of rows) with s_i 1's in i -th column and with different rows. This is the existence problem.

The corresponding optimization problem is in minimization of the number of the possible repeated rows. The problem might be solved, in particular, by algorithms, constructing the matrices in a column-by-column fashion, by partitioning the sets of similar (equal) rows received in a previous step. The first column has been constructed substituting s_1 1's and $m-s_1$ 0's. Without loss of generality we assume that the 1's are substituted on the first s_1 rows. The second column has been constructed by partitioning the intervals (sets of similar rows) of the first column (of lengths s_1 and $m-s_1$) - substituting 1's and 0's on these intervals such that the summary length of one-intervals (where all 1's are substituted), is equal to s_2, and the summary length of zero-intervals (where 0's are substituted), is equal to $m-s_2$. The partitioning of intervals for current k-th column is arbitrary, providing only that the summary length of all one-intervals is equal to s_k. Such construction provides the following property: for each pair of rows, (i,j), where rows i and j belong to different intervals, we have that the i-th and j-th rows are different. Within each interval we have sets of equal rows. The intervals with 1 length in each column don't participate in further partitioning, but they are used (substituting 1 or 0) to provide the summary values s_k and $m-s_k$ on the current k-th column. When in some column there are all 1 length intervals, then all rows are different, and the required matrix is constructed. The remainder columns might be constructed arbitrarily. The graphical scheme is the following:



This is the existence problem as was mentioned. In the similar optimization problem the row repetitions is to be minimized. Let, in a current state of construction we have intervals of lenghts $m_{n_1},\cdots,m_{n_p}$ (greater than 1) on the n-th column. Each of the $m_{n_1},\cdots,m_{n_p}$ intervals consists of the same rows repetitions. The number of pairs of rows - (i,j), where rows i and j are the same, equals $\sum_{j=1}^{p} C^2_{m_{n_j}} = \frac{1}{2}\sum_{j=1}^{p} m_{n_j}(m_{n_j}-1)$, so this is the subject for optimization.

The construction of (0,1)-matrices might be represented by binary trees. We construct a tree T_m with m leaves. The matrix with m rows corresponds to the root vertex. The submatrix with the first s_1 rows from the first step corresponds to the right subtree, and the submatrix with $m-s_1$ rows corresponds to the left subtree, etc. When the current submatrix consists of a single row, we get a leaf. When for any k, $k \le n$, the k-th layer contains leaves only, then the construction is completed. Otherwise as a result of construction on n-th layer we receive a set of subtrees of weights $m_{n_1},\cdots,m_{n_p}$. The constructed trees belong to the class of summary differential balanced trees with summary balances $D_1,\cdots,D_n$, for the given balances $D_i = s_i - (m-s_i), \quad i=1,\cdots,n$.

[S, 1986], [S, 1995] provide an approximation greedy algorithm, which constructs the target (0,1)-matrices in the above described column-by-column fashion of partitioning. The algorithm provides the optimal construction of each column – i.e. the construction, which provides the maximal number of new (i, j) pairs of different rows in each step. It is proven that the optimal construction of each column is provided by partitioning, which distributes the difference $s_k - (m - s_k)$ “homogeneously” on all current non atomic intervals. Returning to the trees terminology, the matrix constructed by the greedy algorithm implies subtrees on each layer of tree, partitioned such that the difference $R_i - L_i = D_i$ is distributed “equally” on all current subtrees.

So this describes the construction of trees in class of summary differential balanced trees providing the local optimum for the functional from the related combinatorial problem of (0,1)-matrices.

A last note. Let the subtrees of i -th layer with weights $m_{i_1}, \cdots, m_{i_p}$ are partitioned into the subtrees with weights $l_{i_1}, r_{i_1}, \cdots, l_{i_p}, r_{i_p}$ correspondingly. We denote $d_{i_1} = r_{i_1} - l_{i_1}, \cdots, d_{i_p} = r_{i_p} - l_{i_p}$. Then the differential balance on i -th layer is equal to $\max\limits_{1 \le j \le p} d_{i_j}$. Since D_i is distributed “equally” by the greedy partition, $\max\limits_{1 \le j \le p} d_{i_j}$ will have the minimum value among all possible partitions. This is the following property: an algorithm, which is locally optimal by means of (0,1)-matrices, is locally optimal also by means of construction of trees of minimal height in class of summary differential balanced trees.

Conclusion

Resuming, - in problem of constructing the summary balanced binary trees with given differential balances of layers, and with height minimization, it is possible to apply the given above combinatorial greedy algorithm, and then the resulting tree has a property that the maximal value of the differential balances on tree layers are optimal - minimal. In terms of search trees this is an extension of perfect balanced trees on layers, when additional constraints are applied.

Bibliography

[K, 1973] D. Knut,.The Art of Computer Programming, vol.3. Sorting and Searching, Addison-Wesley Publishing Company, 1973.

[R, 1977] E. Reingold, J. Nivergelt and N. Deo. Combinatorial Algorithms, Theory and Practice, Prentice-Hall, 1977.

[A, 1989] A. Andersson. Improving Partial Rebuilding by Using Simple Balance Criteria. In Frank Dehne, Jorg-Rudiger Sack, and Nicola Santoro, editors, 1st International Workshop on Algorithms and Data Structures, Lecture Notes in Computer Science, volume 382, pages 393--402. Springer-Verlag, 1989.

[A, 1999] A. Andersson. General Balanced Trees, Journal of Algorithms 30, 1-18, (1999).

[R, 1966] H. J. Ryser. Combinatorial Mathematics, 1966.

[S, 1986] H. A. Sahakyan. Greedy Algorithms for Synthesis of (0,1) Matrices, Doklady Akademii Nauk Arm. SSR, v. 83, 5, pp. 207-209 (1986).

[S, 1995] H. Sahakyan. Hierarchical Procedures with the Additional Constraints, II Russia, with participation of NIS Conference, Pattern Recognition and Image Analysis: New Information Technologies, Ulianovsk, 1995, pp.76-78.

Author information

Hasmik Sahakyan - Institute for Informatics and Automation Problems, NAS Armenia, P.Sevak St. 1, Yerevan-14, Armenia; e-mail: hasmik@ipia.sci.am

Levon Aslanyan – Institute for Informatics and Automation Problems, NAS Armenia, P.Sevak St. 1, Yerevan-14, Armenia; e-mail: lasl@sci.am

DEFINING INTERESTINGNESS FOR ASSOCIATION RULES

T. Brijs, K. Vanhoof, G. Wets

Abstract: Interestingness in Association Rules has been a major topic of research in the past decade. The reason is that the strength of association rules, i.e. its ability to discover ALL patterns given some thresholds on support and confidence, is also its weakness. Indeed, a typical association rules analysis on real data often results in hundreds or thousands of patterns creating a data mining problem of the second order. In other words, it is not straightforward to determine which of those rules are interesting for the end-user. This paper provides an overview of some existing measures of interestingness and we will comment on their properties. In general, interestingness measures can be divided into objective and subjective measures. Objective measures tend to express interestingness by means of statistical or mathematical criteria, whereas subjective measures of interestingness aim at capturing more practical criteria that should be taken into account, such as unexpectedness or actionability of rules. This paper only focusses on objective measures of interestingness.

Keywords: IJ ITA, formatting rules.

Introduction

The problem of finding association rules $X \Rightarrow Y$ was first introduced in 1993 by Agrawal, Imielinski and Swami [1993] as the data mining task of finding frequently co-occurring items in a large Boolean transactional database *D*. Typical applications include retail market basket analysis [Brijs et al., 1999; Brijs et al. 2000], item recommendation systems, cross-selling, loss-leader analysis, etc. In the classical framework, an association rule is considered to be interesting if its support (*s*) and confidence (*c*) exceed some user-defined minimum thresholds. *Support* is defined as the percentage of transactions in the data that contain all items in both the antecedent and the consequent of the rule, i.e. $P(X \cap Y) = \{X \cap Y\}/\{D\}$. *Confidence* on the other hand is an estimate of the conditional probability of *Y* given *X*, i.e. $P(X \cap Y)/P(X)$.

Several authors [Aggarwal & Yu, 1998], however, criticized the use of support and confidence for defining interesting associations. There are several reasons for this. First of all, it is not trivial to set good values for the minimum support and confidence thresholds. Optimally, given unlimited computing resources, these values should be dependent on the size of the data, the sparseness of the data and the particular problem under study. With respect to the size of the data, both the number of rows and columns in the data have an impact on the computing time and the number of association rules being generated. Indeed, for most association rule algorithms, computing time is known to be linear with the number of records in the database [Agrawal & Srikant, 1994]. Furthermore, given a particular percentage threshold for support, the absolute support for a rule will be totally different for a small or a large database. This also has an important impact on the statistical robustness of an association rule and is better known as sampling variability [DuMouchel & Pregibon, 2001; DuMouchel, 1999]. The idea is that association rules with low absolute support should be handled with care since small changes in the absolute support of an association rule with low support have a much greater impact than for an association rule with high absolute support. For example, for a rule with absolute support 2, an absolute support increase of 2 implies the rule to become twice as important as the original rule. In contrast, for a rule with absolute support of 2000, an absolute increase of 2 implies almost no change in the importance of the rule. With respect to the number of columns/items in the data, it is known that this may have a dramatic impact on the computing time, especially if the data is not sparse since the number of potential frequent candidates (and thus also computing time) will increase dramatically with the number of columns in the data. However, as long as the data are sparse, an increase in the number of columns in the database will not significantly increase the computing time due to the clever downward closure principle of frequent itemset mining [Agrawal & Srikant, 1994].

Furthermore, there is a fundamental critique in so far that the same support threshold is being used for rules containing a different number of items. Indeed, intuitively it is not clear why the same support threshold should apply for itemsets of size 2 or of size 7. Clearly, we expect the latter to occur much less frequently so, in some sense, it seems intuitive to specify different (i.e. lower) thresholds for itemsets of increasing size.

Finally, the nature of the problem under study may dictate the support and/or confidence thresholds that should be used. For instance, setting the support threshold too low may lead to rules for which the target group of customers of a particular marketing campaign based on those rules is too small. On the other hand, setting the threshold too high may lead to rules that are trivial for the retailer. Unfortunately, setting the right values for minimum support and confidence today remains to be an unsolved problem in association rule mining.

Another limitation with regard to the support-confidence framework is that high confidence should not be confused with high correlation, neither with causality between the antecedent and the consequent of the rule. The former can be illustrated by the following example.

X	1	1	1	1	0	0	0	0
Y	1	1	0	0	0	0	0	0
Z	0	1	1	1	1	1	1	1

Clearly, from this table it can be observed that *X* and *Y* are positively correlated and that *X* and *Z* are negatively correlated. Yet, if we calculate the support and confidence of the rules $X \Rightarrow Y$ (25%, 50%) and $X \Rightarrow Z$ (37.5%, 75%) it turns out that the second rule dominates the first both in terms of support and confidence. This example demonstrates that one should be very careful in defining the interestingness of rules in terms of support and/or confidence.

A second example illustrates why confidence can be misleading to define interesting rules. Suppose the following situation. Among 5000 customers:

- 3000 buy cola
- 3750 buy cheese
- 2000 both purchase cola and cheese

Then, the rule buy cola ⇒ buy cheese (40%, 66%) could indicate a promising rule. However, although the rule has promising confidence, it is totally misleading since the baseline frequency of customers buying cheese is 75%. In other words, among all customers buying cola, the proportion of customers buying cheese is even lower than in the total group of customers. This example illustrates that one should always take into account the baseline frequency of the consequent of the rule when evaluating the interestingness of association rules.

Interestingness Measures

In the next paragraphs, we will provide an overview of most of the well-known objective interestingness measures, together with their advantages or disadvantages. Furthermore, all measures are symmetric measures, so the direction of the rule ($X \Rightarrow Y$ or $Y \Rightarrow X$) is not taken into account. The reason why we do not discuss a-symmetric measures is that, to our opinion, in retail market basket analysis it does not make sense to account for the direction of a rule since the concept of direction in association rules is meaningless in the context of causality. The interested reader is referred to Tan et al. [2001] for an overview of interestingness measures (both symmetric and a-symmetric) and their properties.

Lift / Interest

A few years after the introduction of association rules, researchers [Aggarwal & Yu, 1998; Brin et al., 1998] started to realize the disadvantages of the confidence measure by not taking into account the baseline frequency of the consequent. Therefore, the lift (also called interest) measure was introduced:

$$I = \frac{P(X \cap Y)}{P(X)P(Y)}$$

Since *P(Y)* appears in the denominator of the interest measure, the interest can be seen as the confidence divided by the baseline frequency of *Y*. The interest measure is defined over $[0, \infty[$ and its interpretation is as follows:

- If $I < 1$, then *X* and *Y* appear less frequently together in the data than expected under the assumption of conditional independence. *X* and *Y* are said to be negatively interdependent.
- If $I = 1$, then *X* and *Y* appear as frequently together as expected under the assumption of conditional independence. *X* and *Y* are said to be independent of each other.
- If $I > 1$, then *X* and *Y* appear more frequently together in the data than expected under the assumption of conditional independence. *X* and *Y* are said to be positively interdependent.

For instance, the interest value for the cola/cheese example equals *I* = 0.4/(0.6*0.75)=0.888 which is clearly below 1 and indicates that buying cola and buying cheese is negatively interdependent, as we expected.

There are, however, two important limitations to the interest measure [DuMouchel & Pregibon, 2001; DuMouchel, 1999]. The first one is related to the problem of sampling variability (see section Empirical Bayes Estimate). This means that for low absolute support values, the value of the interest measure may fluctuate heavily for small changes in the value of the absolute support of a rule. This problem is solved by introducing an Empirical Bayes estimate of the interest measure. The second problem is that the interest measure should not be used to compare the interestingness of itemsets of different size. Indeed, the interest tends to be higher for large itemsets than for small itemsets. The reason is that due to the conditional independence assumption in the denominator of the interest measure, the value in the denominator decreases much more rapidly than the value of the nominator when the number of items in the itemset increase. Therefore, the value of the interest will usually overestimate the interestingness of large itemsets.

Chi-square Test for Independency

A natural way to express the dependence between the antecedent and the consequent of an association rule $X \Rightarrow Y$ is the correlation measure based on the Chi-square test for independence [Brin et al., 1998]. Using the cola/cheese example again, the following contingency table can be derived from it:

	Buy cheese	Do not buy cheese	Total
Buy cola	2000	1000	3000
Do not buy cola	1750	150	2000
Total	3750	1150	5000

The chi-square test for independence is calculated as follows, with O_{xy} the observed frequency in the contingency table and E_{xy} the expected frequency (by multiplying the row and column total divided by the grand total):

$$\chi^2 = \sum_x \sum_y \frac{\left(O_{xy} - E_{xy}\right)^2}{E_{xy}} =$$

$$\frac{\left(2000 - \frac{3000*3750}{5000}\right)^2}{\frac{3000*3750}{5000}} + \frac{\left(1000 - \frac{3000*1150}{5000}\right)^2}{\frac{3000*1150}{5000}} + \frac{\left(1750 - \frac{2000*3750}{5000}\right)^2}{\frac{2000*3750}{5000}} + \frac{\left(150 - \frac{2000*1150}{5000}\right)^2}{\frac{2000*1150}{5000}} =$$

$$417.63 >> 3.84$$

For the *p*-value of 0.05 with one degree of freedom, the cutoff value equals 3.84. Consequently, buying cola and buying cheese can be considered as highly interdependent at the 95% confidence level. The correlation measure therefore seems an attractive alternative to the interest measure. However, the correlation measure has some important limitations with respect to using large data sets [Silverstein et al., 1998].

First of all, the Chi-square test rests on the normal approximation to the Binomial distribution. This approximation

breaks down when the expected values (E_{xy}) are small. More specifically, the Chi-square test should only be used when all cells in the contingency table have expected values greater than 1 and at least 80% of the cells have expected values greater than 5. In market basket data, however, these requirements are easily violated. Secondly, the values in the cell of the contingency table will typically be very unbalanced in the case of association rules. The reason is that the combination of non-existence of the items in the antecedent and consequent is usually much larger than the co-occurrence of its items. In other words, in real applications the upper left cell will be several orders of magnitude smaller than the lower right cell of the contingency table. This situation will usually invalidate the use of the Chi-square test for independence. Finally, the Chi-square test will produce larger values when the data set grows to infinity. Therefore, more items will tend to become significantly interdependent if the size of the dataset increases. The reason is that the Chi-square value depends on the total number of transactions, whereas the critical cutoff value only depends on the degrees of freedom (which is equal to 1 for binary variables) and the desired significance level. Therefore, whilst comparison of Chi-squared values within the same data set may be meaningful, it is certainly not advisable to compare Chi-squared values across different data sets. In any of these cases, an exact test (like Fisher's exact test) to measure the significance of the interdependency is preferred over the Chi-square approximation.

The advantage of the chi-square measure, on the other hand, is that it takes into account all the available information in the data about the occurrence or non-occurrence of combinations of items, whereas the lift/interest measure only measures the co-occurrence of two itemsets, corresponding to the upper left cell in the contingency table.

Correlation Coefficient

The correlation coefficient (also known as the Φ-coefficient) measures the degree of linear interdependency between a pair of random variables. It is defined by the covariance between the two variables divided by their standard deviations:

$$\rho_{XY} = \frac{P(X \cap Y) - P(X)P(Y)}{\sqrt{P(X)(1-P(X))}\sqrt{P(Y)(1-P(Y))}}$$

where ρ_{XY} = 0 when *X* and *Y* are independent and ranges from [-1, +1].

Log-linear Analysis

A natural extension of the Chi-square test of independence between two-way contingency tables is the log-linear analysis [Agresti, 1996]. Log-linear analysis is suited to measure the interdependency between multi-way contigency tables. This kind of test is suited when we are not interested in finding the interdependency between the antecedent and the consequent of an association rule, but we are interested in the interdependency of individual items within an itemset. The log-linear model is one of the specialized cases of generalized linear models for Poisson-distributed data. Log-linear models are commonly used to evaluate multi-way contingency tables that involve three or more variables.

The basic strategy in log-linear modelling involves fitting models to the observed frequencies in the cross-tabulation of categorical variables. The models can then be represented by a set of expected frequencies. Models will vary in terms of the marginals they fit, and can be described in terms of the constraints they impose on the associations or interactions that are present in the data. Once expected frequencies are obtained, different models can be compared that are hierarchical to one another. The purpose is then to choose a preferred model, which is the most parsimonious model that fits the data. The choice of a preferred model is typically based on a formal comparison of goodness-of-fit statistics (likelihood ratio test) associated with models that are related hierarchically (i.e. models containing higher order terms also implicitly include all lower order terms). For instance, the fully-saturated log-linear model for two variables *X* and *Y* is:

$$\ln(F_{ij}) = \mu + \lambda_i^X + \lambda_j^Y + \lambda_{ij}^{XY}$$

where ln(F_{ij}) is the log of the expected cell frequency of the cases for cell *ij* in the contingency table, μ is the overall mean of the natural log of the expected frequencies, λ represents 'effects' which the variables have on the cell frequencies, *X* and *Y* are the variables, and finally *i* and *j* refer to the categories within the variables. The above model is the fully-saturated model since it includes all possible one-way and two-way effects. In order to find a more parsimonious model that will isolate the effects best demonstrating the data patterns, a *non-saturated* model must be discovered. This can best be achieved by setting some of the effect parameters equal to zero. For instance, if the effects parameter λ_{ij}^{XY} is set to zero (i.e. we assume that *X* has no effect on *Y* and vice versa), the following unsaturated model is obtained:

$$\ln(F_{ij}) = \mu + \lambda_i^X + \lambda_j^Y$$

Moreover, the unsaturated and the saturated model are hierarchically related, i.e. they are said to be *nested*. This is a very attractive feature since it validates the use of the likelihood ratio test (LRT). In fact, if F_{ij} represents the observed frequency, and f_{ij} the fitted frequency, then the likelihood ratio test is defined as:

$$LRT = 2\sum_i \sum_j F_{ij} \log\left(\frac{F_{ij}}{f_{ij}}\right)$$

The LRT test is distributed Chi-square with degrees of freedom equal to the number of cells minus the number of non-redundant parameters in the model. In other words, the degrees of freedom equal the number of λ parameters set equal to zero. In fact, the LRT tests the residual frequency not accounted for by the effects in the model. Thus, larger values for LRT indicate that the model does not fit the data well, and thus the model should be rejected. At this point, the LRT can be used to compare the saturated model with any other nested model:

$$LRT_{\text{difference}} = LRT_{\text{nested}} - LRT_{\text{saturated}}$$

with the degrees of freedom equal to the degrees of freedom of the nested model minus the degrees of freedom of the saturated model. If the $LRT_{\text{difference}}$ is not significant, it means that the more parsimonious nested model is not significantly worse than the saturated model. So, then one should choose the nested model since it is simpler (contains less effects).

In some sense, however, the problems associated with Chi-square analysis (see earlier) are also true for the log-linear model. One should therefore be very careful in the interpretation of the results. Again, comparison on the basis of log-linear analysis of the results between different data sets should be avoided.

Empirical Bayes Correction

DuMouchel and Pregibon [2001] and DuMouchel [1999] suggested the use of an Empirical Bayes estimate for the interest value in order to account for the sampling variability in the case of small numbers, which is typically the case when the minimum support threshold specified by the user is small. In that case, slight changes in the absolute value of the support have a large impact on the interest value and this should be corrected for by means of a shrinkage estimate. The procedure goes as follows:

- we have a collection of pairs (n,e), where n is the absolute support of the itemset (above the minimum threshold), and where *e* is the expected absolute support value under the assumption of conditional independence
- for each itemset, *n* is assumed to be drawn from a Poisson distribution with mean $\mu = \lambda * e$.
- however, instead of assuming all λ's to be equal, it is assumed that the λ's are distributed according to a family of prior distributions $\pi(\lambda \mid \theta)$, such as a Gamma distribution or a mixture of Gamma's to have more parameters and thus be more flexible.
- the unconditional distribution for each *n* can now be calculated as $f(n) = \int Poi(n \mid \lambda e)\pi(\lambda \mid \theta)d\lambda$ where *Poi* is the Poisson distribution.
- from the product of this unconditional distribution over all the itemsets, the maximum likelihood estimate $\overset{\frown}{\theta}$ can be calculated.
- Using this maximum likelihood estimate $\overset{\frown}{\theta}$, we can now calculate the posterior density of λ for each pair (*n*, *e*) as $Poi(n \mid \lambda e)\pi(\lambda \mid \overset{\frown}{\theta}) / f(n)$
- the mean of this posterior distribution for each parameter λ will now provide us with an Empirical Bayes shrinkage estimate of the true interest value for each itemset.

The procedure can be easily programmed within the software WinBugs 1.4 and has already been successfully applied on several datasets. On some data sets, however, significant autocorrelations of high-order lags are identified implying the need for rather long chains of iterations which may slow down the calculations significantly.

Nevertheless, empirical results illustrate that the method is indeed able to downsize the interest values for itemsets with low counts. For itemsets with large counts, the Empirical Bayes estimate of the interest does almost not differ from the interest calculated on the raw data. This method is thus clearly preferred over the classical interest measure, especially when the support threshold is being small.

Conclusions

The following suggestions can be formulated based on the analysis of the different interestingness measures discussed in the previous paragraphs:

- Confidence is never the preferred method to compare association rules since it does not account for the baseline frequency of the consequent.
- The lift/interest value corrects for this baseline frequency but when the support threshold is very low, it may be instable due to sampling variability. However, when the data set is very large, even a low percentage support threshold will yield rather large absolute support values. In that case, we do not need to worry too much about sampling variability. A drawback of the interest measure is that it cannot be used to compare itemsets or rules of different size since it tends to overestimate the interestingness for large itemsets.
- Sampling variability can be corrected for by the Empirical Bayes estimate of the interest value. It downsizes the interest when the absolute support of the rule is very low. It generates comparable results to the traditional interest measure when the absolute support is large.
- When association rules need to be compared between data sets of different sizes, the Chi-square test for independence and log-linear analysis are not preferred since they are highly dependent on the dataset size. Both measures tend to overestimate the interestingness of itemsets in large datasets.

Bibliography

[Aggarwal & Yu, 1998] C.C. Aggarwal and P.S. Yu . A New Framework for Item Set Generation. In: Proceedings of the ACM PODS Symposium on Principles of Database Systems, Seattle, Washington (USA), 18-24, 1998.

[Agrawal et al., 1993] R. Agrawal, T. Imielinski and A. Swami A. Mining Association Rules between Sets of Items in Large Databases. In: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington DC (USA), 207-216, 1993.

[Agrawal & Srikant, 1994] R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules. In: Proceedings of the 20th Int'l Conference on Very Large Databases (VLDB), Santiago, Chile, 487-499, 1994.

[Agresti, 1996] A. Agresti. An Introduction to Categorical Data Analysis. Wiley Series in Probability and Statistics, 1996.

[Brijs et al., 1999] T. Brijs, G. Swinnen, K. Vanhoof and G. Wets. The use of association rules for product assortment decisions: a case study. In: Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining, San Diego (USA), August 15-18, 254-260, 1999.

[Brijs et al., 2000] T. Brijs, B. Goethals, G. Swinnen, K. Vanhoof and G. Wets. A data mining framework for optimal product selection in retail supermarket data: the generalized PROFSET model. In: Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining, August 20-23, Boston MA (USA), 300-304, 2000.

[Brin et al., 1998] S. Brin, R. Motwani and C. Silverstein. Beyond Market Baskets: Generalizing Association Rules to Dependence Rules. In: Data Mining and Knowledge Discovery, Vol. 2(1), 39-68, 1998.

[DuMouchel, 1999] W. DuMouchel. Bayesian Data Mining in Large Frequency Tables, With an Application to the FDA Spontaneous Reporting System. In: The American Statistician, Vol. 53, 177-202, 1999.

[DuMouchel & Pregibon, 2001] W. DuMouchel and D. Pregibon. Empirical Bayes Screening for Multi-Item Associations. In: Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco (USA), 67-76, 2001.

[Silverstein et al., 1998] C. Silverstein, R. Motwani and S. Brin. Beyond market baskets: Generalizing association rules to correlations. In: Data Mining and Knowledge Discovery, Vol. 2(1), 39-68, 1998.

[Tan et al., 2002] P-N. Tan, V. Kumar and J. Srivastava, J. Selecting the Right Interestingness Measure for Association Patterns. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton (CA), 32-41, 2002.

Author information

Tom Brijs – Limburgs Universitair Centrum, Research Group Data Analysis and Modelling, Universitaire Campus, gebouw-D, B-3590 Diepenbeek, Belgium; e-mail: tom.brijs@luc.ac.be

Koen Vanhoof - Limburgs Universitair Centrum, Research Group Data Analysis and Modelling, Universitaire Campus, gebouw-D, B-3590 Diepenbeek, Belgium; e-mail: koen.vanhoof@luc.ac.be

Geert Wets - Limburgs Universitair Centrum, Research Group Data Analysis and Modelling, Universitaire Campus, gebouw-D, B-3590 Diepenbeek, Belgium; e-mail: geert.wets@luc.ac.be

THE HOUGH TRANSFORM AND UNCERTAINTY

V.S.Donchenko

Abstract: The paper deals with the generalisations of the Hough Transform making it the mean for analysing uncertainty. Some results related Hough Transform for Euclidean spaces are represented. These latter use the powerful means of the Generalised Inverse for description the Transform by itself as well as its Accumulator Function.

Keywords: Uncertainty, Hough Transform, Accumulator Function, Generalised Inverse.

Introduction

This report is the attempt to represent Hough Transform (HT) [Hough 1962] as a tool for analysis of the uncertainty

Some results, besides, are represented for the vector observations in the scheme of the Hough Transform as well as for complex observations in this case.

The Hough Transform (HT) for well over forty years has been and continues to be an important tool in analysis of shape and pattern recognition. But potentially the Transform seems to be much more than engineering tool only. The idea of the Transform may be used for analyzing uncertainty in much more general cases than in its classical variant. In the paper [Donchenko 1994] the general concept of the Hough Transform within the Hough-pair of the spaces was proposed. Later, in [Donchenko, Kirichenko, 2001] [Donchenko, Kirichenko 2002] the powerful Generalized Inverse apparatus [Nashed, Votruba 1976]], [Albert 1977], [Kirichenko 1997], [Kirichenko, Lepeha 2001] applied to describe HT and Accumulator Function (AF). In the work the results from [[Donchenko, Kirichenko, 2001] and [Donchenko 1999] are extended on the vector case.

General concept of the Hough Transform – Hough-pare of spaces

General concept of the HT[Donchenko 1994] as a tool for analysis of the uncertainty may be built on the base of so called Hough-pare of spaces, which are virtually a pare of sets S_o, S_p enhanced by its subsets G_θ L_s: $G_\theta \subseteq S_o$, $L_s \subseteq S_p$, $\theta \in S_p$, $s \in S_o$, mutually indexed. One set (space) S_o is interpreted as a space of observations, another S_p – as a space of parameters. This space of parameters is interpreted as a variety of the variants for uncertainty which corresponds to observation s.

These subsets are agreed in the next sense: for any pare (s,θ) $\theta \in L_s \Rightarrow s \in G_\theta$ and conversely: $s \in G_\theta \Rightarrow \theta \in L_s$. Some additional conditions may be added to these: about type of interception for example.

The HT within the Hough-pare is determined as a transition from the observation s – or its sequence s_1, $s_2,...,s_N \in S_o$ – to subset L_S , correspondingly –sequence of subsets $L_{s_1} = L_1$, $L_{s_2} = L_2,...,$ $L_{s_N} = L_N$ – of another space, indexed by the observations.

Such determination permits the description of straight and inversed HT, Hough-estimator, Fast HT as a sequential HT and Fast HT as the HT of the complex observations.

Each of the observations is supposed to be taken by the choice of the parameter $\theta_1,..., \theta_N$ and the consecutive choice of the elements s_1, $s_2,...,s_N \in S_o$ from correspondent subsets of G_θ -type. Besides, the observations may be disturbed in that or this way. In this case one says that the elements are observed with an error.

The parameters $\theta_1,..., \theta_N$ –some of them may be equal - is said to be represented in s_1, $s_2,...,s_N \in S_o$.

The set of parameters represented in the sample is supposed to be “comparatively small” and the main target of the HT-based analysis is ascertainment – estimation - of that set. Properly, this is the task of the estimation in the theory of the HT.

HT may be applied to the sequence s_1, $s_2,...,s_N \in S_o$ taken without previous choice of , $\theta_1,..., \theta_N$. In that case HT-analysis is targeted to describe “comparatively small” set of the parameters, “concentrated” the observations. This task may be called the task of the clustering in HT.

As to complex observations S, then we say it to be the subset S = $S_{i_1,...,i_k}$ = {s_{i_1}, s_{i_2},..., s_{i_k}}, {i_1, i_2,..., i_k}$\subseteq${1,…,N} of the initial observations. And by HT of the complex observation S= $S_{i_1,...,i_k}$ we will call the set L_S= $L_{i_1,...,i_k}$, determined by the relation:

$$L_S= L_{i_1,...,i_k} = \bigcap_{j=1}^{k} L_{i_j} .$$

Sometimes HT of the complex observations is called Fast HT. Fast HT can cut essentially the set of the parameters, pretended to be represented in the sample.

This variant of the Fast HT one ought to differ from another using, when the Accumulator Function (AF) of the HT consecutively calculated for some set C - rough approximation - and its consecutive – detailed - partitions.

AF is determined by the HT of the sample – original or complex observations – as a function of set C in the space of parameter in one of the next two senses: absolute A(C) or relative NA(C) –by the relations:

$$A(C) = \Sigma_{i=1}^{N} \delta(C\cap L_i), , C\subseteq S_p , \quad (1)$$

$$NA(C)= A(C)/N =N^{-1} \Sigma_{i=1}^{N} \delta(C\cap L_i),.C\subseteq S_p , \quad (2)$$

where $\delta(C)$, $C \subseteq S_p$, equal 1, if C is not empty and 0, if C – empty set.

The summing in (1), (2) for the complex observations is by the set of the complex observations under consideration.

Argument C in the AF depends on the concrete types of spaces. For the Euclidean spaces for parameters and observations set C may be: ball as in (10), (11) below; hyper-cube; compact and so on.

AF is the mean to estimate the set of parameter, which are represented in the sample or the "smallest" set of the parameters in the clustering task. Properly, such set (or sets) is the set of maximum for AF.

Hough Transform in the Euclidean spaces

In its original variant HT was determined for the case, when

- S_o, S_p are appropriate rectangles in R^2;
- parametric sets G_θ, $\theta=(\rho,\varphi)\in R^2$ is the set of the graphics of the straight lines in the normal representation: $G_\theta= G_{(\rho,\varphi)} =\{(x,y)\in R^2: \rho = x\cdot\cos\varphi +y\cdot\sin\varphi\}$;
- parametric set $L_s=L_{(x,y)}$, $s=(x,y)\in R^2$ is the set of parameters for which correspondent lines G_θ include observation s=(x,y): $L_s=L_{(x,y)}= \{(\rho,\varphi)\in R^2: \rho = x\cdot\cos\varphi +y\cdot\sin\varphi\}$.;

Observations s=(x,y) may be with an error so without it. In the first case $y = \bar{y} +\varepsilon_{(x,y)}$, where $\varepsilon_{(x,y)}$- the error of an observation. Errors, which correspond to different observations, are supposed to be independent but not obligatory identically distributed.

One of the generalizations of that original variant may be such one, in which the spaces in the Hough-pare are any Euclidean spaces or their appropriate subsets:

- $S_o=R^m$, $S_p =R^n$
- G_θ, $\theta\in S_p$, is determined the graphic of mappings $y=g(x,\theta)$, $\theta\in R^l$ from R^n in R^m . The sample s_1, s_2,..., s_N consists of the pares $s_i=(x_i, y_i)$:

$$y_i = g(x_i,\theta_i)\in R^m, x_i \in R^n , \quad (3)$$

$$y_i = g(x_i,\theta_i)+ \varepsilon_i \in R^m, x_i \in R^n , i=1,…,N \quad (4)$$

Variant (4) represents the scheme of observations with an error, (3) – without it.

HT for such sample is the sequence L_1, L_2,..., L_N of the subsets from R^l, where

$$L_i=\{ \theta\in R^l: y_i = g(x_i,\theta)\}, I=1,…,N.$$

Particularly, if the set of mapping is of affine-type (linear + shift) from R^n in R^m, then the matrix $\theta=A\in R^{m\times(n+1)}$ of this map may be considered as the parameter, i.e. l= m×(n+1) .

HT, AF and Hough-estimator are described on the sample (x_i,y_i), i=1,N, $x\in R^n$, $y\in R^m$ points of the graphics of the affine set of the mappings $y = A \begin{pmatrix} x \\ 1 \end{pmatrix}$, - $x\in R^n$, $y\in R^m$, A –m×(n+1)) matrix, $\begin{pmatrix} x \\ 1 \end{pmatrix}$– block vector-column from $x\in R^n$ and 1.

These observations may be observed in the scheme without the error (3) or with it (4), correspondingly:

$$y_i = A_i \begin{pmatrix} x_i \\ 1 \end{pmatrix}, \quad (5)$$

$y_i = A_i\begin{pmatrix}x_i\\1\end{pmatrix}+\varepsilon_{x_i}$, $x_i\in R^n$, $y_i\in R^m$, $A_i\in R^{m\times(n+1)}$, i =1,…,N. (6)

As it was remarked earlier specific parameter $A_i\in R^{m\times(n+1)}$, i =1,…,N corresponds to each of the observations. Only scheme with the error (6) and independent errors will be considered below. The last means, that errors of the observations ε_{x_i} , i =1,…,N are independent. The distribution of ε_x will be denoted by P_x:

$P_x(B^{(m)})= P\{\varepsilon_x \in B^{(m)}\}$, (7)

$B^{(m)}$ – Borel set from R^m.

HT $L_{(x,y)}$ of an observation (x,y) is the set of affine transforms, mapping x in the observed y, which may be disturbed:

$L_{(x,y)} = \{A\in R^{m\times(n+1)}: y = A\begin{pmatrix}x\\1\end{pmatrix}\}$. (8)

$S_r(\theta)$ below denotes the r-ball with a center in the θ in the space of all m×(n+1) matrixes with the trace norm, induced by the trace scalar product:

$(A,B) = \mathrm{tr}\, A'B = \Sigma_i (A'B)_{ii} =\Sigma_{ij} a_{ij}b_{ij}$.

The trace norm, obviously, coincides with the Euclidean norm in $R^{m\times(n+1)}$.

AF in absolute or frequency variants will be defined for the balls $S_r(\theta)$ as for arguments and denoted correspondingly $A_r(B)$, $NA_r(B)$:

$A_r(B) =A(S_r(B))= \Sigma_{i=1}^N \delta(S_r(B)\cap L_i)$, , (9)

$NA_r(B)= A_r(B)/N =N^{-1} \Sigma_{i=1}^N \delta(S_r(B)\cap L_i)$, $B \in R^{m\times(n+1)}$. (10)

Theorem 1. AF for the sample (x_i,y_i), i=1,N points of affine observations may be represented by next expression:

$A_r(B) = \Sigma_{i=1}^N \delta(S_r(B)\cap L_i) = \sum_{i=1}^N \delta(\varepsilon_{x_i} \in S_{r\sqrt{1+||x|_i|^2}}((B-A_i)\begin{pmatrix}x_i\\1\end{pmatrix})$. (11)

Proof. Accordingly with the theorem 2 [2]

$A_r(B) = \Sigma_{i=1}^N \delta(S_r(B)\cap L_i) = \Sigma_{i=1}^N \delta(||y_i - B\begin{pmatrix}x_i\\1\end{pmatrix}||^2\le r^2(1 + ||x_i||^2)$, $B \in R^{m.\times(n+1)}$. (12)

As for the each of observations $y_i = A_i\begin{pmatrix}x_i\\1\end{pmatrix}+\varepsilon_{x_i}$, i =1,…,N,

then condition $||y_i - B\begin{pmatrix}x_i\\1\end{pmatrix}||^2 \le r^2(1 + ||x_i ||^2)$

in (12) is equivalent to the condition $|| A_i\begin{pmatrix}x_i\\1\end{pmatrix}+\varepsilon_{x_i} - B\begin{pmatrix}x_i\\1\end{pmatrix}||\le r\sqrt{1+|| x_i ||^2}$,

that proves the theorem.

Remark 1. $S_r(B)$ in (12) is the r-ball in the trace norm in the matrix space with the center in a B, then the $S_{r\sqrt{1+||x_i||^2}}((B-A_i)\begin{pmatrix}x_i\\1\end{pmatrix})$ is the $r\sqrt{1+|| x_i ||^2}$ -ball in R^m with the center in $(B-A_i)\begin{pmatrix}x_i\\1\end{pmatrix}$, i=1,…,N.

Corollary 1. Obviously, $\delta(S_r(B)\cap L_i)$, i =1,…,N are Bernoulli-distributed random variables with the parameters, determined by the expressions

$p_i = P\{\varepsilon_{x_i} \in S_{r\sqrt{1+||x||^2}}((B-A_i)\begin{pmatrix}x_i\\1\end{pmatrix}\}$, i=1,…,N. (13)

Proof. The result is the consequence of taking 1 for each of $\delta(S_r(B)\cap L_i)$, i =1,…,N in (12).

$\delta(\varepsilon_{x_i} \in S_{r\sqrt{1+||x||^2}}((B-A_0)\begin{pmatrix}x_i\\1\end{pmatrix})$, i=1,…,N.

Theorem 2.(0-1 Law). The limit value of the AF when with probability 1 is finite or infinite as $n\to\infty$. It is finite iff $\sum_{n=1}^{\infty} p_i = \sum_{n=1}^{\infty} P\{\varepsilon_{x_i} \in S_{r\sqrt{1+||x||^2}}((B-A_i)\begin{pmatrix}x_i\\1\end{pmatrix}\} < \infty$

Proof. The proof repeats that one for scalar case in [8].

Theorem 3. The next limit take place with the probability 1:
$\lim_{N\to\infty}(N^{-1}\sum_{i=1}^{N}\delta(S_r(B)\cap L_i)) - N^{-1}\sum_{i=1}^{N}p_i) = \lim_{N\to\infty}(NA_r(B)) - N^{-1}\sum_{i=1}^{N}p_i) = 0$, where p_i, i=1,…,N are determined by (13).
Proof. As in was in previous case the proof repeats that one for scalar case in [8].

Remark 2. For the case under consideration – vector case – all the consequences from [8] for scalar observations are valid.

Remark 3. The statements of the theorems earlier are free from constraints on the distribution of an error. Besides, the distribution may depends on x.

Theorem 4. AF for the sequence of K complex observations may be represented by next expression:

$$A_r(B) = \Sigma_{i=1}^{K}\delta(S_r(B)\cap \text{Ł}_i) = \Sigma_{i=1}^{K}\delta(\ ||(Y_i - BX_i)X_i^{+}|| \le r)\ ,\ B \in R^{m\times(n+1)},$$

where:

- Ł_i, i=1,…,K – Hough transforms for the complex observations,
- A^+- General Inverse for A,
- Y_i, i=1,...,K – block-matrixes from the y-components of the original observations the complex observation consists of,
- X_i, i=1,…,K– block-matrixes from the x-components of the original observations the complex observation consists of,

Conclusion

The subject matter of the paper is generalizing the Hough Transform that convert it in mathematical tool with wide range of application in analyzing the "uncertainty". The abstract form of the HT is represented within the framework of Hough-pare of spaces.

The HT for observations and parameters from Euclidean spaces has been represented and investigated for affine sets of transforms. The author would like to believe that the results represented are the only one step to promote HT to be the mean for uncertainty analyzing.

Bibliography

[Albert 1977]Albert A. Regression, pseudo inverse, recurrent estimation..–M.: Nauka, 1977 г., 305 p(in Russian).

[Donchenko 1994] Donchenko V.S. General Scheme of the Hough Transform and properties of the Hough estimates in special case of Discrete Spaces.//Statist. Res. Rep.–Umea, Sweden: University of Umea,1994.–1994-6, S-901 87. 10 p.

[Donchenko 1999] Donchenko V.S. Limit theorems for Accumulator Function in Hough Transform Scheme. // Bulletin of the University of Kiev. Series: Physics \& Mathematics. –1999, Vol..№ 1, cc.191-195.(In Ukrainian)

Donchenko, Kirichenko,2001] Donchenko V.S., Kirichenko M.F. Hough Transform and Generalized Inverse. // Bulletin of the University of Kiev, Series: Physics \& Mathematics.– 2001, vol. № 4, , pp.191-196. (in Ukrainian)

[Donchenko , Kirichenko 2002] Donchenko V.S., Kirichenko M.F. Fast Hough Transform and Generalized Inverse.// Journal of Automation and Information Sciences . – 2002, vol. №2, pp..115-125. (In Russian, also translated in English)

[Hough 1962] Hough P.V.C. Method and Means for Recognizing Complex Patterns. - U.S. Patent 3069354, 1962.

[Kirichenko 1997] Kirichenko M.F. Analytical representation for Generalized Inverse disturbances // Cybernetics and System Analysis. – 1997, №2, pp.98-107. (In Russian)

[Kirichenko, Lepeha 2001]. Kirichenko M.F., Lepeha M.P. Disturbances for pseudo-inverse and projective matrixes and their applications to linear and non-linear identification.// Journal of Automation and Information Sciences. - 2001, №1, c.6-23. (In Russian, also translated in English)

[Nashed, Votruba 1976] Nashed M. Zuhair ,Votruba G.F. A Unified Operator Theory of Generalized Inverse, Proceedings of an Advanced Seminar Sponsored by the Mathematical Research Center, The University of Wisconsin, Madison, October 8-10, 1973. – New York, Academic Press, 1976.

Author information

Volodymyr S. Donchenko – Kyiv National Taras Shevchenko Universiry, Professor.
2, Akademician Glushkov prospectus, building 6, Kyiv,03680, Ukraine; e-mail: vsdon@unicyb.kiev.ua

SYSTEMS ANALYSIS: THE STRUCTURE-AND-PURPOSE APPROACH BASED ON LOGIC-LINGUISTIC FORMALYZATION

Lyudmila M. Lukiyanova

Abstract: Systems analysis (SA) is widely used in complex and vague problem solving. Initial stages of SA are analysis of problems and purposes to obtain problems/purposes of smaller complexity and vagueness that are combined into hierarchical structures of problems(SP)/purposes(PS). Managers have to be sure the PS and the purpose realizing system (PRS) that can achieve the PS-purposes are adequate to the problem to be solved. However, usually SP/PS are not substantiated well enough, because their development is based on a collective expertise in which logic of natural language and expert estimation methods are used. That is why scientific foundations of SA are not supposed to have been completely formed. The structure-and-purpose approach to SA based on a logic-and-linguistic simulation of problems/purposes analysis is a step towards formalization of the initial stages of SA to improve adequacy of their results, and also towards increasing quality of SA as a whole. Managers of industrial organizing systems using the approach eliminate logical errors in SP/PS at early stages of planning and so they will be able to find better decisions of complex and vague problems.

Keywords: industrial organizing system, problem situation, systems analysis, quality of systems analysis, purpose structures correctness, structure-and-purpose approach, situations control, logic-and-linguistic simulation, analytic evaluation.

Introduction

We consider here industrial organizing systems. Along with such general characteristics as uniqueness, unpredictable behaviour in concrete situations, capacity to adapt to changing environmental conditions, and to alter a structure, industrial organizing systems possess several particularities. Their structures are usually hierarchical. Their purposes depend on social and other factors. Their production and technological equipment are standardized and unified to a great degree. Volume and assortment of their products are changing dynamically. Their level of technological renewal is very high. These and some other factors give rise to complex problems that are often characterized by a high level of vagueness. Therefore SA becomes a necessary means of efficient function and development for the majority of industrial organizing systems.

Initial stages of SA (analysis of problems/purposes and synthesis hierarchical SP/PS) are based on expert knowledge and experience. Experts elicit problems/purposes, determine the main problem/purpose, decompose it to create the structure of problems/purposes of smaller complexity and vagueness, estimate PS-purposes to combine them into a final PS. All these tasks are creative, but determining the main problem/purpose, decomposing them to obtain the ones of smaller and smaller complexity, greater and greater certainty, and then combine them into hierarchical SP/PS are more crucial because a lot of informal factors have to be allowed. Tthe majority methods of decomposing purposes are based on a collective expertise in which the logic of the natural language is used. Experts use subjective models and collective interpretation of purposes, achieving which allows the managerial staff (MS) to solve problems. Natural language logic and high level of subjectivism very often stipulate the results that are not logically valid. Logical errors in a PS very often cause insufficient solving of the main problem and sometimes even failure in solving that problem.

Widely used approaches to systems analysis stem from the PATTERN method [1] and among the methods of decomposing purposes to reduce their complexity and vagueness the methods proposed in [Черняк, 1975], [Поспелов, Ириков, 1986], [Перегудов, Тарасенко, 1989], [Saaty, Kearns, 1991], [Силич, Тарасенко, 1982], [Кондратов, Ростанец, 1982], [Романов, Клыков, 1974] are more useful for industrial organizing systems. However, together with their evident value for theory and practice of SA these and the majority of other well known methods and approaches have two essential drawbacks: the requirements to the wording of the purposes (problems, criteria, functions) have not been clearly defined, rules of decomposition of the purposes to form hierarchical PS have been formulated in a general way. It means that the methods are not constructive enough. Well-constructed methods have been also developed ([Nilson, 1973], [Романов, Клыков, 1974], etc.), but the method [Nilson, 1973] and similar methods are effective for closed and not so large worlds. The method [Романов, Клыков, 1974] and similar methods can be used for open and large worlds, but the majority of syntactically correct decisions automatically generated by means of them are semantically meaningless, pragmatically useless and therefore inefficient. Besides, conformities to natural laws, and principles of problems/purposes setting in industry have not been investigated enough. That is why scientific foundations of SA of industrial organizing systems cannot be considered completely formed.

So, the following *SA*-scientific-and-technical problem may be formulated for industrial organizing systems: there is *no objective, constructive to a great degree and integral approach to systems analysis that could guarantee the development of correct SP/PS and defining efficient PRS which are adequate to the main problem solved.* Namely because of the SA-problem, MS often create a contradictory, incomplete SP/PS and an inadequate PRS. This fact, resulting in logical errors in a PS, usually reveals itself only in the course of achieving the purposes, hampers determining SP/PS-substantiation and creating a sufficient enough PRS as the system that can achieve the main purpose and solve the main problem of an organizing system. So, it also reduces SA-quality as a whole.

To solve the SA-scientific-and-technical problem it is necessary *to determine basis concepts and establish conformities to natural laws of purpose-setting* (the first sub-problem of SA-problem), *to explore semantics of problem/purpose formulations and relations between them in SP/PS* that so far have been declared in a natural language and have not been studied enough (the second sub-problem), *to determine strict to the wording SP/PS-properties such as discrepancy and completeness because* so far they have also been declared in the natural language, consequently they are usually polysemantic and not clear enough (the third sub-problem of SA-problem). Therefore integral, the more objective and constructive SA-methodology has to be investigated.

Basis concepts of problem/purpose-setting

SA involves two main processes with inter-reverse time motion: purpose-setting (p-setting) in which a desired result of activity is being formulated and purpose-achieving (p-achieving) in which a real result of activity is being achieved by means of the PRS. Here we determine basis concepts of problem/purpose-setting for industrial organizing systems such as a problem, a need, a purpose, a SP/PS (all concepts of SA-semantic field are considered in detail in [Lukiyanova, 2002]).

A *need* in something is always objective. If a need cannot be satisfied simply, it is a reason of the problem situation in the organizing system. *A problem* is contradiction between something desired and something that is being (e.g. between a desired situation and a current situation; it means that a current situation needs correction).

A purpose is always subjective. We consider concept 'purpose' as a general name to designate a desired result of activity that is used in p-setting to characterize MS-desire, and a desired result of activity that is used in p-achieving to characterize MS-ability in his own system. Analysis of different definitions of 'purpose' allowed us to formulate the following generalized definition of this concept:

<a purpose> ::= < a desired result of activity > [<a structure >] [<time>]. (1)

The definition (1) consists of three semantic multipliers. The two last semantic multipliers marked by square brackets are facultative. Actually, not each purpose is considered as a structure (e.g. a simple purpose; for its achieving MS know and has the means). The second facultative semantic multiplier usually characterizes a task of purpose achieving (time is one of redistributed resources for purpose achieving).

Analysis of more than thousand formulations of problems and purposes showed that semantics of connections between formulations of problems and purposes that solve the problems are very close, and purposes of industrial organizing systems are often like negations of the problems, and according to (1) purposes may be simple or complex. Analysis also showed that relations between PS-problems and between SP-purposes are usually identical.

A structure of the main, complex and vague, problem/purpose is a SP/PS in which problems/purposes are combined by means of structural relations such as subordination, compared, completeness, and the other relations that are used to evaluating of SP/PS-correctness or allow for resources that MS has to purpose achieving [Lukiyanova, 2002].

Conformities to natural laws of purpose-setting in industry

Problems hamper function and development of a system and can stipulate needs in something. A simple problem stipulates a simple need. A simple need can stipulate a simple purpose:

$$\text{a need} \xrightarrow{\text{motivation}} \text{a purpose.} \qquad (2)$$

The formula (2) expresses *the first law of purpose-setting*. For a simple purpose, p-setting according to (2) is completed if MS has the means to p-achieving:

$$\text{a purpose} \xrightarrow{\text{means}} \text{a result.} \qquad (3)$$

A new problem is not arisen in this case. But p-setting is continuing if MS has not the means to p-achieving. Therefore t*he second law of purpose-setting is:*

$$\text{a desired result} \rightarrow \text{desired means.} \qquad (4)$$

Complex problem is a reason of a complicated purpose. A complicated purpose is considered as a system and its structure is analyzed. Formula (4) defines *a basis strategy of PS-creating. A few additional strategies of PS-creating* are considered in [Лукьянова, 2001]. Besides, purposes descriptions and possibilities of its decomposing in organizing systems are studied. The following aspects of purposes in their formulations are defined: rational limiting of redundancy and significance; evident (in contrast to implicit) semantics of purposes descriptions of an industrial organizing systems that expresses purpose parts of their natural language formulations, functions of the parts, basis elements of the systems and their determination in space of properties. We use the following possibilities of complicated purposes decomposing: status (external-internal, etc.), aspects of activities (social, economical, control, industrial, etc.), kinds of economical activity (in according to the classificatory of the economical activity kinds), control functions, kinds of industrial activity, etc.

P-setting in any industrial organizing system orients on an external purpose (purposes) established by the above-system. Therefore we suggest that external purposes express absolute value for the subordinate organizing systems and purposes that are set in the analyzed system as sub-purposes of the external purpose express utilitarian value. Thus in PS-creating every purpose excepting the main purpose and purposes- leaves of PS may be considered as absolute value to the subordinate purpose and as utilitarian value to the above-purpose. This is *the third law of purpose-setting*.

Structure-and-purpose analysis of industrial organizing systems

We suggest a new 'structure-and-purpose' paradigm of SA in industrial organizing systems. It uses two determining concepts of purpose systems ('a structure' and 'a purpose') and allows for purpose domination in SA. Actually, an organizing system is a means of its complex purpose achieving. Therefore it is useful to analysis the structure of the purpose (a purpose may be considered as a problem negation) to reduce its complexness and vagueness. Criteria and functions of PRS are also semantically connected (it is used criterion form of a purpose description and there are functional properties in purpose formulations [Lukiyanova, 2002]). Besides, purposes (and PS) determine PRS itself and dominate as in p-setting as in p-achieving processes. Therefore for industrial organizing systems we postulate the following.

Postulate 1. Abstract semantics of purposes and relations between the purposes determines logical models of problems and purposes analysis and hierarchical SP and PS as the results of these process.

The postulate is based on the hierarchical structures of the industrial organizing systems, on the roles in activities of different parts of such systems, and on the concept of absolute and utilitarian value of the parts.

Postulate 2. Formal logic-semantic analysis of problems/purposes will able to help MS to obtain discrepancy and completeness SP/PS.

To orient in variety methods, techniques, procedures of problems/purposes analysis-synthesis of SP/PS and investigate a method using of which can guarantee against logical errors in SP/PS we classified, as shown on the figure 1, all possible methods including well-known and wide used ones. We used informal, partial-formal and formal degrees of describing their following aspects (bases of the classification): a *purpose description language* to realize input interface between the experts/MS and the formal logic-semantic system (the first level of the classification), *rules of decomposition of purposes* that check correctness of PS-creating (the second level of the classification), *means of description of PS and its characteristics* to realize output interface between the formal logic-semantic system and the experts/MS (the third level of the classification). The empty classes of methods are shown on the figure 1 as black circles.

Among the classes, we note the following three homogeneous ones: K^{111} involves informal methods, K^{222} involves partially formal methods, K^{333} involves formal methods. Methods of the rest classes are inhomogeneous.

So, *the first realized class is K^{111}*. It involves the most number of the methods such as [Лопухин, 1971], [Черняк, 1975], [Перегудов, Тарасенко, 1989], [Saaty, Kearns, 1991] and similar ones. Advantages of the methods of this class are a universal and an all-round analysis (i.e. purpose decomposition and at the same time PS-estimation). The main disadvantage of these methods is polysemy of purposes, rules of decomposition of purposes, and means of description of a PS and its characteristics. Polysemy causes high-level subjectivism of analysis and hampers finding logical errors in a PS.

The second realized class is K^{113}. It involves methods that are similar the method [Поспелов, Ириков, 1986]. Advantage of these methods is a formal description of a PS, but this ability does not increase a constructive level of the methods to reduce logical errors in the PS.

The third realized class is K^{221}. It involves methods that are equivalent the method [Силич, Тарасенко, 1982]. They standardize purpose descriptions in a very strict form and PS-description in a scenario form. The methods allow decomposing purposes automatically at some steps of PS-forming. The main drawbacks of these methods are impossibility decomposing new purposes (problems) and using the methods for other organizing systems. *The fourth realized class is K^{231}*. It involves methods like the method [Романов, Клыков, 1974] which is well-constructive and flexible at the same time. However, the automatically formed PS may be semantically meaningless and pragmatically useless. Besides, the methods do not define PS-characteristics by means of which it can be checked logical validation of PS.

The fifth realized class is K^{311}. It involves methods that are equivalent the method [Кондратов, Ростанец, 1982]. These methods used formal purpose descriptions that are faintly connected with decomposing possibilities.

The sixth realized class is K^{331}. It involves methods that are similar the method [Nilson, 1973]. As a rule, these methods work in closed and not so large worlds. The other their disadvantage is impossibility of allowing for semantics of purposes, relations between them, and PS-properties.

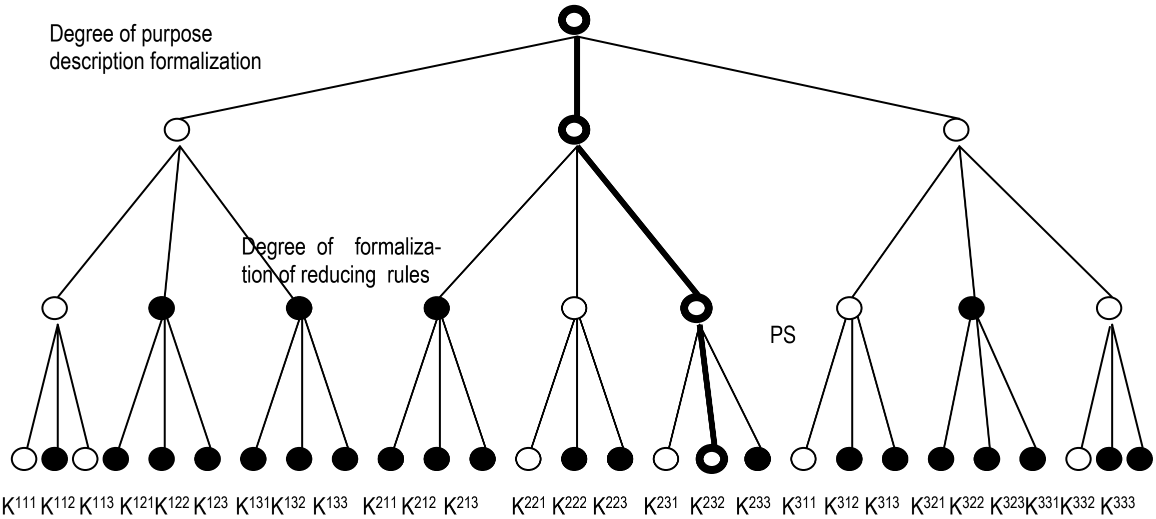

Degree of purpose description formalization
Degree of formalization of reducing rules
PS
K^{111} K^{112} K^{113} K^{121} K^{122} K^{123} K^{131} K^{132} K^{133} K^{211} K^{212} K^{213} K^{221} K^{222} K^{223} K^{231} K^{232} K^{233} K^{311} K^{312} K^{313} K^{321} K^{322} K^{323} K^{331} K^{332} K^{333}

Figure 1. The classification of methods of creating hierarchical PS

The classification helped us to find the more adequate class of methods than classes used so far, to set and solve the SA-scientific-and-technical problem of industrial organizing systems. It is a class K^{232} which characteristics marked out on the figure 1 by the bold line. Methods of this class use a necessary (for perception) level of partial-formal description of problems/purposes, partial-formal description of SP/PS, and formal rules of problems/purposes analysis. Man-machine systems based on such methods have the following advantages: 1) problems/purposes descriptions in a constraint natural language are effective as for experts/MS as for formal systems that analyze problems/purposes, and realize intellectual interface between them; 2) a logic model of problems/purposes analysis based on problem area semantics does possible eliciting errors of problems/purposes structure analysis; 3) graphic SP/PS imaginations together with problems/purposes descriptions in a constraint natural language are effective to realize output interface.

According to characteristics of class K^{232} and the postulates 1 and 2, the following principles of structure-and-purpose analysis (considered in detail in [Lukiyanova, 2002]) are established:

1. It is inter-reverse time motion logical causality between p-setting and p-achieving processes.
2. Man-machine structural analysis of problems/purposes (criteria, functions of PRS) as adequate practical reasoning (*in contrast to inadequate man reasoning or decomposing problems/purposes algorithm*) is expedient.
3. Hierarchical system structures such as SP and PS, PS and structure of functions (SF) of PRS, SF of PRS and structure of PRS, etc. are connected semantically and logically.

4. Current situations in industrial organizing systems stipulate problems/purposes (criteria, functions of PRS) analysis.
   4.1. Formalization of problems/purposes analysis must allow for semantics of problems/purposes and relations between them (*logical stratum*):
      4.1.1. Partial linguistic formalization of problems/purposes (criteria, functions of PRS) provides evident expressing its semantics.
      4.1.2. Logic-semantic formalization of analysis of problems/purposes provides logical discrepancy, model completeness of conclusion and semantic applicability of inference rules.
         4.1.2.1. Discrepancy of a hierarchical SP/PS is stipulated by the principle 3.1.2.
         4.1.2.2. Completeness of a hierarchical SP/PS is stipulated by the principle 3.1.2.
      4.1.3. Classification of purpose situations simplifies selection of current analyzing strategy.
      4.1.4. Partial graph-and-linguistic formalization of SP/PS (structure of criteria (SC), SF of PRS) provides adequate imagination of structural analysis results.
   4.2. Formalization of p-achieving estimation provides allowing for resources of p-achieving (*mathematic stratum*).
5. Narrow-minded perception principle (facultative principle).

To realize principle 4 we suggest a conceptual model problems/purposes (criteria, functions of PRS) analysis using the fundamental idea of the situation control theory [Поспелов, 1995]. Let us consider the conceptual model shown on figure 2. In accordance with the 4.1-4.2-principles of structure-and-purpose analysis of problem situations in industrial organizing systems, it is stratified into two stratums: logical and mathematical. The logical stratum of SP/PS-creating uses logical methods to check correctness of SP/PS. The mathematical stratum uses mathematical methods to problems/purposes estimating.

In accordance with the 4.1.1-principle, intellectual interface language (L_{in}^1) of structure-and-purpose analysis to describe problems/purposes (criteria, functions of PRS) is suggested. As it is studied the most suitable kind of a language to describe problems and purposes is the frame language [Лукьянова, 2001] that is based on the two-level linguistic model of problem/purpose. The first level (macro-describer) is a role frame expressing a functional formula of activity in an industrial organizing system. The second level (micro-describer) is a describer of functional elements in space of their properties. The space of properties is divided into some groups. By means of the groups it is ordered in-role problem/purpose description by means of specific terms. Each group of properties determines its own way of problems/purposes decomposition. Because of natural language redundancy the problem/purpose parts of problem/purpose formulations are marked by special pointers. The roles, the kinds of properties and a problem/purpose pointer (H/G) express external semantics of problems/purposes. Internal semantics of problems/purpose is expressed by basis concepts that form terms. It is also developed the language L_{in}^2 to realize input interface with basis knowledge. It is developed as simplified version of L_{in}^1.

According to the 4.1.2-principle logic-semantic formalization of problems/purposes analysis is investigated [Lukiyanova, 2002]. It is used the semiotic model theory [Осипов, 1995] and the logic of utilitarian values [Ивин, 1970]. As it is shown on figure 2, the three-components semiotic system consists of a formal subsystem S_T, Ψ-re-constructor that reconstructs S_T in accordance with the current situation in the bush of problems/purposes, O-reformer that reforms the current linguistic representation of problem/purpose into a logical formula and vice versa:

$$O: \ p \ = \ (H/G)\, f_j [\, [\, [\, H/G\,]\, f_s\,] \dots] \leftrightarrow \left\{ \begin{array}{l} (H/G)\, f_j [\, [\, \wedge [H/G\,]\, f_r] \dots] [\, \supset f_7\,] \\ \\ \quad [\, f_s [\, [\, \wedge f_r\,] \dots] \supset]\, (H/G)\, f_{7.} \end{array} \right\} \qquad (5)$$

Here p – a linguistic representation of a problem/purpose (alternatives of a linguistic representation of a problem/purpose are involved in figured brackets);
f – a role phrase in p (j, r, s={1, 2, …, 6}).

According to the 4.1.3-principle it is classified situations on purposes [Lukiyanova, 2002]. Six classes are defined, but the only one is correct. Discrepancy and completeness of the hierarchical SP/PS are also defined.

In accordance with the 4.1.4-principle of structure-and-purpose approach to SA, the partial-formal structural language (L_{out}) to realize output interface of the SP-s and SG-s bases is suggested. It is used a theory-graphic tree-model which nodes are described in L_{in}^1 and theory-set language to describe (semantically) complicated arcs [Lukiyanova, 2002]. The semiotic system via Intellectual interface takes away experts/MS linguistic descriptions of problems/purposes, reforms them by means of O-reformer into logical formulae and checks correctness of a current bush of the SP/PS-problems/purposes by means of its logical subsystem S_T. Logical subsystem S_T uses adequate fragments of basis knowledge of the problem situation as its own domains by means of Ψ-re-constructor in the time of checking the bush of problems/purposes. If the current bush of problems/purposes of the SP/PS is not valid, S_T identifies the logical error and forms recommendation to correct the problem/purpose of the bush of the SP/PS. The semiotic system and knowledge base of problem situation are realized in Delphi.

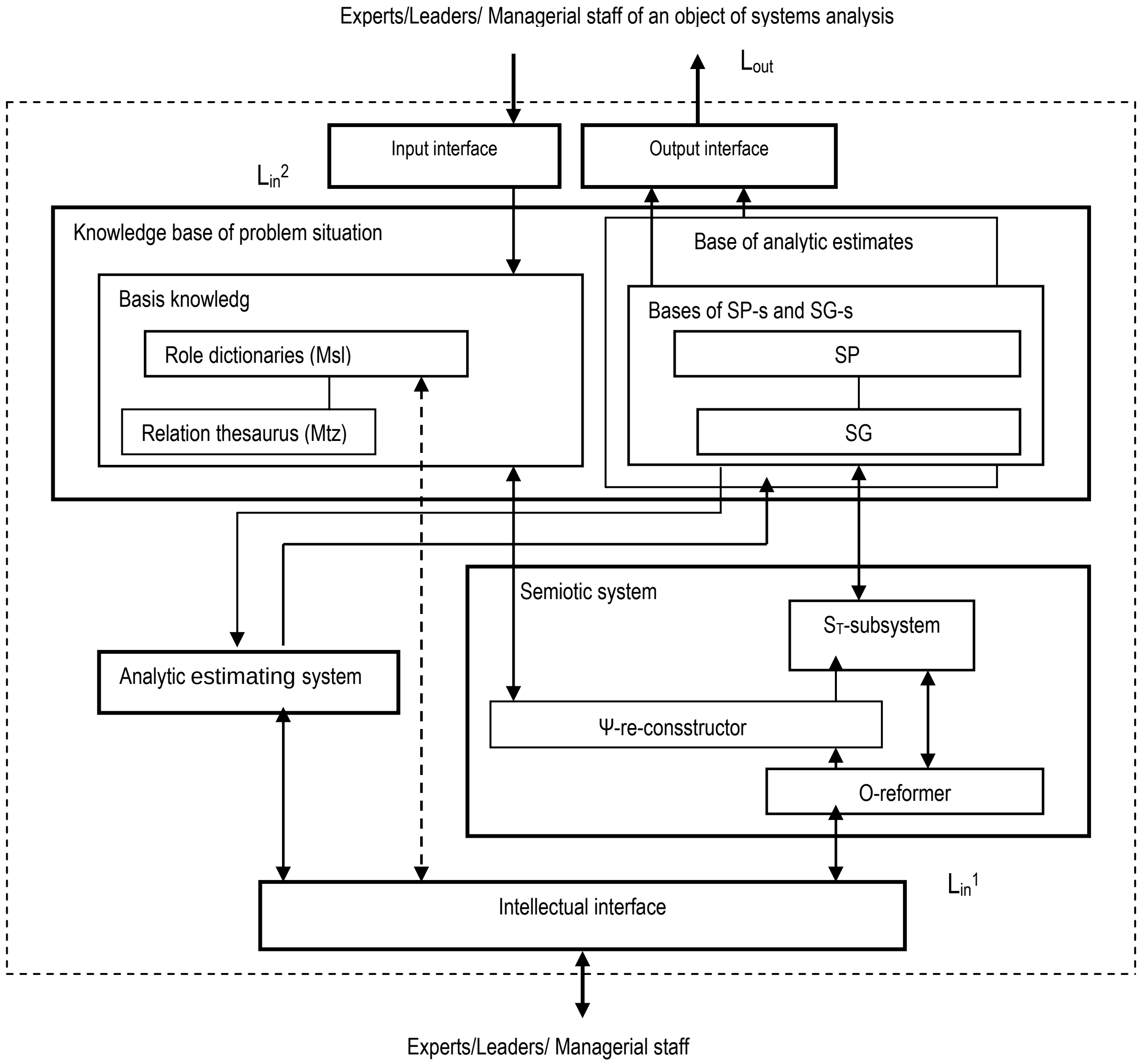


Figure 2. The conceptual model of problems/purposes analysis

According to the 4.1.4-principle an analytic subsystem realizing the analytic hierarchy process [Saaty, Kearns, 1991] can be used as to the separate problem/purpose as to the bush of problems/purposes estimating, and even to the SP/PS estimating as a whole.

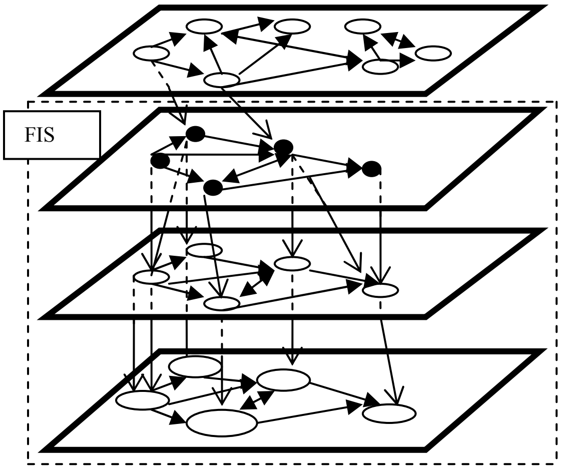
In the time of analysing the bush of problems/purposes of the SP/PS S_T is invariable and works by steps. One step is an inference that produces in according with a scheme: $p_1 |\Rightarrow p_2$ (p_1 and p_2 are problems/purposes) in which semantic relations are conditions of inference rule applicability. In contrast to traditional relations semantic ones are determined as $< l_j, R_j >$ in which the first component (l_j,) is a relation name, $l_j \in I$, I is a set of names expressing relative basis Mtz of a problem field [Lukiyanova, 2002]. An inference consists of the following acts: for $< p_1, p_2 >$ it is hypothesized implicative connection $p_1 \rightarrow p_2$ in which p_1 supposed as truth and a corresponding purpose as an absolutely valuable; truth meaning of $p_1 \rightarrow p_2$ is estimated by basis knowledge (fig. 2) and if it is truth then in according with modus ponens p_2 is supposed as truth. If in according with basis knowledge truth meaning of $p_1 \rightarrow p_2$ is false it is identified as contradiction and S_T inferences permissible p_2'. An inference is simplified by classification of situations on $< p_1, p_2 >$ and $< p_1, p_2, p_i., \ldots, p_n >$. Analysis of criteria of purposes achieving is based on the PS. The main criterion usually corresponds to the main purpose and local criteria correspond to local purposes of the PS. Analysis of functions of the PRS is also based on the PS. MS determines function for every purpose in the PS. Thus, a SF is formed. The partial-formal method forming two- or three-levels organizing structure of PRS is suggested in [Лукьянова, 2001], [Lukiyanova, 2002]. It consists of systemizing the list of SF-functions and based on the following characteristics grouping: subject-object, control levels and phases, character of production process and life cycle of production. Systemizing leads to determine functions of the control subsystem and the controllable subsystem. Then according to generally accepted rules and norms functions into the subsystems are grouped. The results of analysis are a base to PRS organizing structure synthesis.

Conclusion

The new structure-and-purpose approach to SA is suggested. It covers all stages of SA, makes it possible to systematize as SA-procedures as its results. P-setting laws, problem situation basis knowledge, control of problem solving by means of purposes situation classification, partial-formal imagination of problems/purposes/criteria/functions, its structures, logical-linguistic formalization of a structural SA are established.

The approach is used in fishery industry systems (FIS). Several complex program [Лукьянова, 1986], problem situations in technological equipment designing [Лукьянова, 1988] and in region FIS were analyzed.

So, there were elicited 43 problems in a FIS. At preliminary problems analysis each of problems was analyzed semantically. Also degree of uncertainty and complexity of the problems were fixed: status, aspects and kinds of economical and industrial activities, control functions. Preliminary analysis changed content of some problems and their number (50). Further systematization of problems list showed that the percent of external problems is 10.5%, the majority of internal problems are problems of control (55.5% from a general number of internal problems) and economic problems (26.5%). There are many financial problems (13.5%) among economic ones; organizing (22.5%), planning problems (9.5%) and analyzing problems (9%) among control ones. Systematization of problems stipulates their correct stratification and more exact determination of expert groups. Simplified result of problem stratification is shown on Figure 3.



Social stratum:
social indices (e.g. necessary value of producing fishing production).

Economic stratum:
economic relations and indices (e.g. minimal profit from produced fishing production).

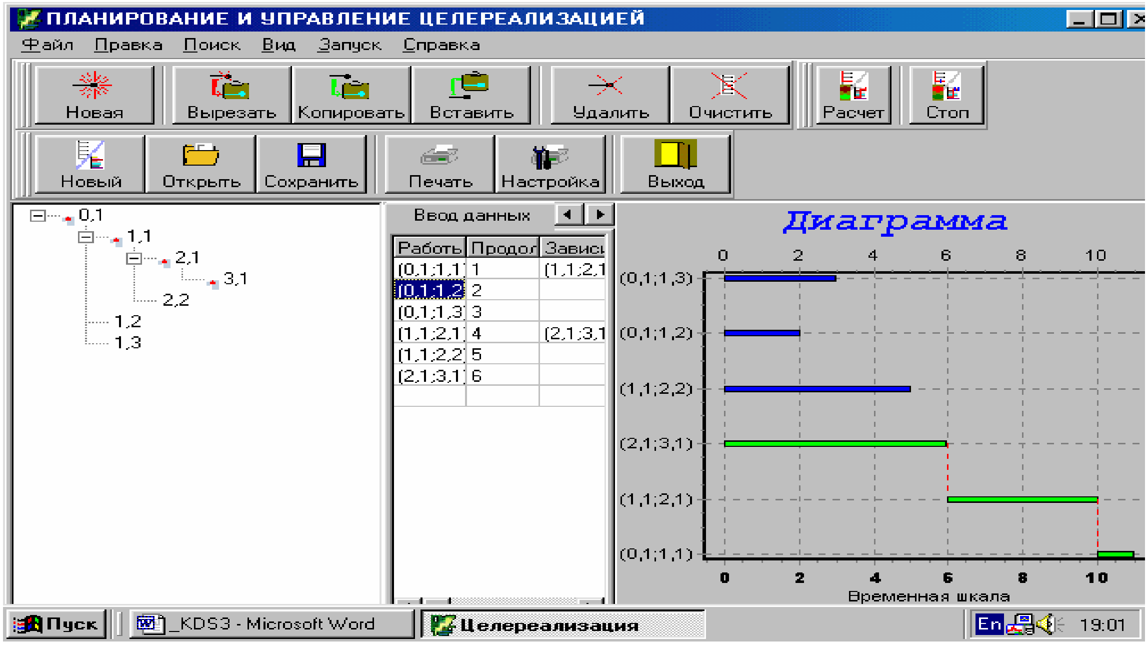
Control stratum:
control relations and parameters (e.g. reduce of demurrage of equipment).

Industrial stratum:
industrial relations and indices (e.g. value of produced fishing production).

Figure 3. The example of the complex problem decomposed

Then cause-and-effective connections into each aspect, kind of economical and industrial activity, control function were analyzed. This analysis helped us to define the main problem which further analyzing was given the correct SP. Analogically the PS was created. Then the PRS was simulated and logically valid line diagram of p-achieving was formed. Example of the intermediate line diagram realized in Delphi is shown on Figure 4. The line diagram based on the PS as a result of p-setting obtained by means of logical-linguistic simulation.

Correctness of the structure-and-purpose approach to SA for FIS is confirmed by problem solving practice. Experts agreed with all logical errors in PS and SP that the semiotic system found and with all recommendations for their correction that the semiotic system formed.



Работь	Продол	Зависі
(0,1;1,1)	1	(1,1;2,1
(0,1;1,2	2	
(0,1;1,3)	3	
(1,1;2,1)	4	(2,1;3,1
(1,1;2,2)	5	
(2,1;3,1)	6	

Figure 4. The example of the intermediate line diagram designed in according with fragment of the PS

Bibliography

[Лопухин, 1971] М.М. Лопухин. ПАТТЕРН – метод планирования и прогнозирования научных работ. М., 1971.

[Черняк, 1975] Ю.И. Черняк. Системный анализ в управлении экономикой. Экономика, М.,1975.

[Поспелов, Ириков, 1986] Г.С. Поспелов, В.А. Ириков. Программно-целевое планирование и управление. М.,1983.

[Перегудов, Тарасенко, 1989] Ф.И. Перегудов, Ф.П. Тарасенко. Введение в системный анализ. Высшая шк., М., 1989.

[Saaty, Kearns, 1991] Thomas L. Saaty, Kevin R. Kearns. Analytical planning: The organization of systems. N. J., 1991.

[Силич, Тарасенко, 1982] В.А. Силич, В.П. Тарасенко. Алгоритмизация и автоматизация процесса построения содержательной модели организационной системы и формирования на ее основе дерева целей. В кн.: Вопросы кибернетики. Методы и модели оценки развивающихся систем, с. 14-35. ВИНИТИ, М., 1982.

[Кондратов, Ростанец, 1982] В.А. Кондратов, В.Г. Ростанец. Применение программно-целевого метода в планировании развития науки и техники. В кн.: Совершенствование показателей плана, с. 153-165. НИИ план-я и нормативов, М., 1982.

[Романов, Клыков, 1974] В.Г. Романов, Ю.И. Клыков. Формирование дерева целей в системах ситуационного управления. Изв. АН СССР: Технич. кибернетика, № 5, с. 11-15,1974.

[Nilson, 1973] Nilson N. Problem Solving Methods in Artificial Intelligence. 1973.

[Поспелов, 1995] Д.А. Поспелов. Ситуационное управление. Новый виток развития. Изв. РАН: Теория и системы управления, № 5, с. 152-158, 1995.

[Осипов, 1981-1982] Г.С. Осипов. Две задачи теории семиотических моделей управления: I. Представления семиотических моделей. Изв. АН СССР: Технич. кибернетика, № 6, с. 100-110, 1981; II. Семиотический анализ. Изв. АН СССР: Технич. кибернетика, № 1, с. 131-137, 1982.

[Ивин, 1970] А.А. Ивин. Основания логики оценок. МГУ, М., 1970.

[Лукьянова, 1986] Л.М. Лукьянова. Метод структурирования целей (на примере структур целей для целевых программ). Изв. АН СССР: Технич. кибернетика, №3, с. 65-75, 1986.

[Лукьянова, 1988] Л.М.Лукьянова. Моделирование анализа проблем проектирования технологического оборудования. В сб.:Тезисы докл. Всесоюзн. конф. по искусств-му интеллекту, Т.1, с. 195-200. ВИНИТИ, М., 1988.

[Лукьянова, 2001] Л.М. Лукьянова. Система поддержки структурно-целевого анализа проблемных ситуаций. В сб.: Труды Междун. научно-практич. конф. KDS-2001, Т. I, с. 446-453. СЗГЗТУ, С.-Петербург, 2001.

[Lukiyanova, 2002] L.M. Lukiyanova. Methodology of Structure-aimed Analysis of Problem Situations in Organization Systems for Industry Branches. SPIIRAS Proceedings. Issue 1, v. 1, p. 297-315. SPIIRAS, SPb, 2002.

Author information

Lyudmila M. Lukiyanova – KSTU, Associate Professor, Active member of International Information Academy;
236000, Kaliningrad, Sovietsky pr, 1, KSTU, Department of Control Systems and Computer Science, Russia;
e-mail: llm_llm@mail15.ru

ADMISSIBLE SUBSTITUTIONS IN SEQUENT CALCULI

A. V. Lyaletski

Abstract: For first-order classical logic a new notion of admissible substitution is defined. This notion allows optimizing the procedure of the application of quantifier rules when logical inference search is made in sequent calculi. Our objective is to show that such a computer-oriented sequent technique may be created that does not require a preliminary skolemization of initial formulas and that is efficiently comparable with methods exploiting the skolemization. Some results on its soundness and completeness are given.

Keywords: completeness, first-order logic, quantifier rule, sequent calculus, skolemization, soundness

Introduction

Investigations in computer-oriented reasoning gave rise to the appearance of various methods for the proof search in the classical 1st order logic. Particularly, sequent calculi were suggested by Gentzen [1]. But their practical application as a logical technique (without preliminary skolemization) of the intelligent systems has not received wide use: preference is usually given to the resolution-type methods. This is explained by higher efficiency of the resolution-type methods as compared to sequent calculi, which is mainly connected with different possible orders of the quantifier rule applications in sequent calculi while resolution-type methods, due to skolemization, are free from this deficiency.

In its turn, the deduction process in sequent calculi reflects sufficiently well natural theorem-proving methods which, as a rule, do not include preliminary formula skolemization so that reasonings are performed within the scope of the signature of the initial theory. This feature of sequent calculi becomes important when some interactive mode of proof is developed since it is preferable to present the output information concerning the proof search in the form usual for man. That is now the problem of the efficient quantifier manipulation makes its appearance.

When quantifier rules are applied, some substitution of selected terms for variables is made. To do this step of deduction sound, certain restrictions are put on the substitution. The substitution, satisfying these restrictions, is said to be admissible. Here we investigate the classical notion of admissible substitution and show how it can be modified so that efficient sequent calculi can be finally obtained. We use the calculus G [2] for the demonstration of the way of the construction of such a modification denoted by mG here. Note that when constructing mG, we don't touch upon any procedure of selection of propositional rules and terms substituted, focussing our attention on quantifier handling only.

Genzen's Notion of Admissible Substitutions

Classical quantifier rules, substituting arbitrary structure terms when applied “from bottom to top", are usually of the following form [2]:

$$\frac{\Gamma_1, A[t/x],\ \forall xA, \Gamma_2 \to \Gamma_3}{\Gamma_1, \forall xA, \Gamma_2 \to \Gamma_3} \qquad (\forall\text{: left})$$

$$\frac{\Gamma_1 \to \Gamma_2, A[t/x], \exists xA, \Gamma_3}{\Gamma_1 \to \Gamma_2, \exists xA, \Gamma_3} \qquad (\exists\text{: right})$$

where the term t is required to be free for the variable x in the formula A. This restriction of the substitution of t for x gives Gentzen's (classical) notion of an admissible substitution, which proves to be sufficient for the needs of the proof theory. But it becomes useless from the point of view of efficiency of computer-oriented theorem-proving methods. It is clear from the following example.

Consider a sequent $A_1, A_2 \to B$, where A_1 is $\forall x_1\exists y_1(R_1(x_1) \vee R_2(y_1))$, A_2 is $\forall x_2\exists y_2(R_1(y_2) \vee R_2(x_2))$, and B is $\exists x_3\forall y_3(R_2(x_3) \vee R_3(y_3))$. The provability of this sequent in calculus G will is established below, while here we notice that quantifier rules must be applied to all the quantifiers occurring in A_1, A_2, and B. Therefore, classical notion of admissible substitution yields 90 (= 6!/(2!*2!*2!)) different orders of the quantifier rule applications ("from bottom to top") to the sequent $A_1, A_2 \to B$. It is clear that resolution type methods allow avoiding this redundant work.

Kanger's Notion of the Admissible Substitutions

To optimize procedure of the applications of quantifier rules, S.Kanger suggested in [2] his calculus of Gentzen type, denoted here by K. In calculus K a "pattern" of a deduction tree is first constructed with the help of special variables, the so called parameters and dummies. At some times an attempt is made to convert a "pattern" into proof tree to complete the deduction process. In case of failure, the process is continued.

The main difference between K and G consists in a special modification of the above quantifier rules and in a certain splitting (in K) of the process of the "pattern" construction into stages. In K the rules ($\forall$: left) and ($\exists$: right) are of the following form:

$$\frac{\Gamma_1, A[d/x], \forall x A, \Gamma_2 \to \Gamma_3}{\Gamma_1, \forall x A, \Gamma_2 \to \Gamma_3} \quad d/t_1,...,t_n$$

$$\frac{\Gamma_1 \to \Gamma_2, A[d/x], \exists x A, \Gamma_3}{\Gamma_1 \to \Gamma_2, \exists x A, \Gamma_3} \quad d/t_1,...,t_n$$

where $t_1,...,t_n$ are the terms occurring in the conclusion of the rules, d is the dummy, and $d/t_1,...,t_n$ denotes that when an attempt is made to convert "pattern" into proof tree, the dummy d must be replaced by one of the terms $t_1,...,t_n$. The replacement of dummies by terms is made in the end of every stage, and at every stage the rules are applied in a certain order.

This scheme of the deduction construction in calculus K leads to a notion of the Kanger-admissible substitution, which is more efficient than the classical one. Thus in the above example it yields only 6 (=3!) variants of different possible orders of the quantifier rule applications (but none of these variants is preferable). Despite this, the Kanger-admissible substitutions still did not allow to attain the efficiency comparable with that when the skolemization is made. It is due to the fact that, as in case of the classical admissible substitution, it is required to select a certain order of the quantifier rule applications when an input sequent is deduced, and, if it proves to be unsuccessful, the other order of applications is tried, and so on.

New Notion of Admissible Substitutions

For constructing the modification mG of calculus G from [2], let us introduce a new notion of admissible substitutions in order to get rid of the dependence of the deduction efficiency in sequent calculi on different possible orders of quantifier rule applications. The main idea is to determine, proceeding from quantifier structures of formulas of an input sequent and a substitution under consideration, would there exists a sequence of desired quantifier rule application. (This notion was used in slightly modified form in [3].)

Substitution is defined as a finite (maybe, empty) set of ordered pairs, every of which contains a variable and a term and is written in the form t/x, where x is the variable and t is the term of substitution [4].

We assume that besides usual variables there are two countable sets of special variables, namely of parameters and dummies.

Let P be a set of sequences of parameters and dummies, and s be a substitution. Put T(P,s) = {<z,t,p>: z is the variable of s, t is the term of s,p P, and z lies in p to the left of some parameter from t}. The substitution s is said to be admissible for P if and only if (1) the variables of s are only dummies and (2) in T(P,s) there are no elements $<z_1,t_1,p_1>$,..., $<z_n,t_n,p_n>$ such that $t_2/z_1 \in s,...,t_n/z_{(n-1)} \in s$, $t_1/z_n \in s$ (n>0).

Calculus mG

As in the case of calculus G, its modification mG deals with formulas, except that in mG every formula from a sequent has a certain sequence of parameters and dummies. Therefore, it is convenient to define calculus mG by means of the pairs <p,A>, where A is the formula and p - the sequence (word) of parameters and dummies. Also, it will be assumed that the empty sequence is always added to all formulas from the input sequent (that is, from the sequent to be proved).

The rules of the calculus mG are the following.

Propositional rules:

$$\frac{\Gamma_1, <p,A>, <p,B>, \Gamma_2 \to \Gamma_3}{\Gamma_1, <p,A \wedge B>, \Gamma_2 \to \Gamma_3} \qquad \frac{\Gamma_1 \to \Gamma_2, <p,A>, \Gamma_3 \quad \Gamma_1 \to \Gamma_2, <p,B>, \Gamma_3}{\Gamma_1 \to \Gamma_2, <p,A \wedge B>, \Gamma_3}$$

$$\frac{\Gamma_1,<p,A>, \Gamma_2 \to \Gamma_3 \quad \Gamma_1 \to \Gamma_2,<p,B>,\Gamma_3}{\Gamma_1,<p,A \vee B>,\Gamma_2 \to \Gamma_3} \qquad \frac{\Gamma_1 \to \Gamma_3,<p,A>,<p,B>,\Gamma_2}{\Gamma_1 \to \Gamma_2,<p,A \vee B>,\Gamma_3}$$

$$\frac{\Gamma_1, \Gamma_2 \to <p,A> \quad <p,B>,\Gamma_1,\Gamma_2 \to \Gamma_3}{\Gamma_1,<p,A \supset B>,\Gamma_2 \to \Gamma_3} \qquad \frac{<p,A>,\Gamma_1 \to \Gamma_3,<p,B>,\Gamma_2}{\Gamma_1 \to \Gamma_2,<p,A \supset B>,\Gamma_3}$$

$$\frac{\Gamma_1, \Gamma_2 \to <p,A>,\Gamma_3}{\Gamma_1,<p,\neg A>,\Gamma_2 \to \Gamma_3} \qquad \frac{<p,A>,\Gamma_1 \to \Gamma_2,\Gamma_3}{\Gamma_1 \to \Gamma_2,<p,\neg A>,\Gamma_3}$$

Quantifier rules:

$$\frac{\Gamma_1, <pd,A[d/x],>,<p,\forall xA>,\Gamma_2 \to \Gamma_3}{\Gamma_1,<p,\forall xA>,\Gamma_2 \to \Gamma_3} \qquad (\forall\text{: left'})$$

$$\frac{\Gamma_1, \to \Gamma_2,<pd,A[d/x]>,<p,\exists xA>, \Gamma_3}{\Gamma_1, \to \Gamma_2,<p,\exists xA>, \Gamma_3} \qquad (\exists\text{: right'})$$

$$\frac{\Gamma_1 \to \Gamma_2,<pz,A[z/x]>, \Gamma_3}{\Gamma_1 \to \Gamma_2,<p,\forall xA>,\Gamma_3} \qquad (\forall\text{: right'})$$

$$\frac{\Gamma_1, <pz,A[z/x]>,\Gamma_2 \to \Gamma_3}{\Gamma_1,<p,\exists xA>,\Gamma_2 \to \Gamma_3} \qquad (\exists\text{: left'})$$

Here d is a new dummy, z is a new parameter, p is a sequence of parameters and dummies, Γ_1, Γ_2, and Γ_3 are arbitrary sequences of pairs, consisting of sequences (of dummies and parameters) and formulas, A, B are arbitrary formulas.

Applying first rules "from bottom to top" to the input sequent and afterwards to its "heirs", and so on, we finally obtain a so-called deduction tree.

A deduction tree D is called a proof tree for the input sequent (in mG) if and only if there exists a substitution of terms for variables, s, such that (1) s is admissible for set of all sequences of parameters and dummies from D and (2) after application of s to the formulas from all upper sequents of D we obtain axioms, that is, the sequents $\Gamma_1 \to \Gamma_2$ such that Γ_1 and Γ_2 contain a common formula.

The main result concerning the calculus mG is as follows.

Theorem. Let $A_1,...,A_m,B_1,...,B_n$ be the formulas of the 1st order language. There exists a proof tree for the input sequent $<,A_1>,...,<,A_m> \to <,B_1>,...,<,B_n>$ in calculus mG if and only if there exists a proof tree for the input sequent $A_1,...,A_m \to B_1,...,B_n$ in calculus G.

Proof.

(=>) Let D be a proof tree for the input sequent $<,A_1>,...,<,A_m> \to <,B_1>,...,<,B_n>$ in the calculus mG, and s be a substitution, which converts all upper sequents of D into axioms and is admissible for set P of all sequences of parameters and dummies from D. Without any loss of generality, we may assume that terms of s do not contain dummies for otherwise these dummies could be replaced by a constant, say, c0.

Since s is admissible for P, it is possible to construct the following sequence p consisting of parameters and dummies which form the sequences of P:

(i) every p' P is a subsequence of p, and

(ii) the substitution s is admissible for {p} (i.e. there is no an element <z,t,p> T({p},s) such that $t/z \in$ s.

Such a sequence p may be generated, for example, by the convolution algorithm from [3], applied to a list of all the sequences from P provided that in the convolution algorithm are treated parameters as existence quantifiers, and dummies universal quantifiers.

Property (i) of the sequence p and formulation of the propositional and quantifier rules permit to make the following assumption:

When D was constructed, propopositional and quantifier rules were applied ("from bottom to top") in the order that corresponds to looking through p from the left to right: i.e. when the first quantifier rule was applied, the first variable (a parameter or a dummy) of p was generated, when the second quantifier rule was applied, the second variable of p was generated, and so on.

Now it is possible to covert the tree D into proof tree D' for the input sequent $A_1,...,A_m \to B_1,...,B_n$ in calculus G. To do this, let us "repeat" the process of the construction of D in the above order p and execute the following transformations:

1) Suppose that in a processed node of D one of the following rules was applied:

$$\frac{\Gamma_1, <pd,A[d/x],>,<p,\forall xA>,\Gamma_2 \to \Gamma_3}{\Gamma_1,<p,\forall xA>,\Gamma_2 \to \Gamma_3} \qquad (\forall\text{: left'})$$

or

$$\frac{\Gamma_1, \to \Gamma_2,<pd,A[d/x]>,<p,\exists xA>, \Gamma_3}{\Gamma_1, \to \Gamma_2,<p,\exists xA>, \Gamma_3} \qquad (\exists\text{: right'})$$

and t/d s for some term t. The term t is free for d in A, because the order of applications of quantifier rules is reflected by p, and property (ii) is satisfied. Therefore, the admissibility in the classical sense will be observed when the above rules (∀: left') and (∃: right') are replaced in D by rules (∀: left) and (∃: right) of the calculus G: and all other occurrences of d in D are replaced by t.

$$\frac{\Gamma_1, A[t/x],\ \forall xA, \Gamma_2 \to \Gamma_3}{\Gamma_1,\forall xA, \Gamma_2 \to \Gamma_3} \qquad (\forall\text{: left})$$

or

$$\frac{\Gamma_1 \to \Gamma_2, A[t/x], \exists xA, \Gamma_3}{\Gamma_1 \to \Gamma_2, \exists xA,\Gamma_3} \qquad (\exists\text{: right})$$

2) In other cases the rules of the calculus mG are replaced by their analogs from G by a simple deleting of sequences of parameters and dummies from these rules.

It is evident that D' is a deduction tree in the calculus G. Furthermore, the way of conversion of D into D' allows making the conclusion that upper sequents of D' are axioms of the calculus G. Thus, D' is a proof tree for the input sequent $A_1,...,A_m \to B_1,...,B_n$ in G.

(<=) Let D' be a proof tree for the input sequent $A_1,...,A_m \to B_1,...,B_n$ in G. Convert D' into tree D, which, as be can seen bellow, is a proof tree for the input sequent $<,A_1>,...,<,A_m> \to <,B_1>,...,<,B_n>$ in mG. For this purpose "repeat" ("from bottom to top") a process of construction of D', replacing in D' every rule application by its analog in mG and subsequently generating substitution s. (Initially s is the empty substitution.)

1) If an applied rule is one of the following:

$$\frac{\Gamma_1, A[t/x],\ \forall xA, \Gamma_2 \to \Gamma_3}{\Gamma_1,\forall xA, \Gamma_2 \to \Gamma_3} \qquad (\forall\text{: left})$$

or

$$\frac{\Gamma_1 \to \Gamma_2, A[t/x], \exists xA, \Gamma_3}{\Gamma_1 \to \Gamma_2, \exists xA,\Gamma_3} \qquad (\exists\text{: right})$$

then it is replaced by

$$\frac{\Gamma_1, <pd,A[d/x],>,<p,\forall xA>,\Gamma_2 \to \Gamma_3}{\Gamma_1,<p,\forall xA>,\Gamma_2 \to \Gamma_3} \qquad (\forall\text{: left'})$$

or

$$\frac{\Gamma_1, \to \Gamma_2,<pd,A[d/x]>,<p,\exists xA>, \Gamma_3}{\Gamma_1, \to \Gamma_2,<p,\exists xA>, \Gamma_3} \qquad (\exists\text{: right'})$$

accordingly with adding t/d to the existing substitution s, where d is a new dummy, and with substituting d for those occurrences of t into "heirs" of the formula A[t/x], which appeared as a result of applying of a replaced rule "inserting" the term t.

2) In all other cases replacement of the rules of G by the rules of mG is evident. (Note that $<,A_1>,...,<,A_m> \to <,B_1>,...,<,B_n>$ is declared as input sequent of D. The rules ($\exists$: left) and ($\forall$: right) may be considered as those inserting new parameters).

Since D' is a proof tree in the calculus utilizing the classical notion of admissible substitution, then it is clear that the finally generated substitution s is admissible (in the new sense) for a set of all sequences of parameters and dummies from D. Therefore, D is a proof tree for the input sequent $<,A_1>,...,<,A_m> \to <,B_1>,...,<,B_n>$ in mG. *Q.E.D.*

Corollary 1. For any formulas $A_1 ,..., A_m, B_1 ,..., B_n$ the formula $(A_1 \wedge ... \wedge A_m) \supset (B_1 \vee ... \vee B_n)$ is valid if and only if there exists a proof tree for the input sequent $<,A_1>,...,<,A_m> \to <,B_1>,...,<,B_n>$ in calculus mG.

Proof.

In accordance with [2] the formula $(A_1 \wedge ... \wedge A_m) \supset (B_1 \vee ... \vee B_n)$ is valid if and only if there exists a proof tree for the input sequent $A_1,...,A_m \to B_1,...,B_n$ in the calculus G. On the basis of the Theorem the latter condition holds true if and only if a proof tree for the input sequent $<,A_1>,...,<,A_m> \to <,B_1>,...,<,B_n>$ can be constructed in calculus mG. *Q.E.D.*

To demonstrate the deduction technique, consider the sequent $A_1,A_2 \to B$ from the above example and establish its provability in calculus G. To do this, construct a proof tree for the input sequent $<,A_1>,<,A_2> \to <,B>$ in calculus mG and use the Theorem.

Applying to the initial sequent only quantifier rules we can receive the following sequent:

$<d_1z_1,R_1(d_1) \vee R2(z_1)>,<,A_1>,<d_2z_2,R_1(z_2) \vee R_2(d_2)>,<,A_2> \to <d_3z_3,R_2(d_3) \vee R_3(x_3)>, <d_3z_3, R_2(d_3) \vee R_3(x_3)>,<,B>$, where $d_1,...,d_4$ are dummies, $z_1,...,z_4$ are parameters.

Now let us apply prorositional rules to the last sequent as long as they are applicable. As a result, we get a deduction tree D. If we generate the substitution $s = \{z_2/d_1, z_3/d_2, c_0/d_3, z_1/d_4\}$ (c_0 is a constant), then we can draw the following conclusions concerning s and D:

1) s is admissible for the set of all sequences of dummies and parameters from D, and

2) every upper sequent from D may be transformed into axioms by applying of s to it.

So, in accordance with the above Theorem the sequent $A_1,A_2 \to B$ is provable in the calculus G. *Q.E.D.*

Some Reconstruction of mG

The formulation of the calculus mG shows that the order of the quantifier rule applications is immaterial. In the calculus mG the quantifier rules are needed to determine a quantifier structure of formulas from the input sequent. This observation gives us possibility to construct a modification mG' of the calculus mG, which contains the so-called doubling rules instead of all the quantifier rules.

Doubling rules:

$$\frac{\Gamma_1,<pdz_1...z_k,A>,<pd'u_1...u_k,A[d'/d,u_1/z_1,...,u_k/z_k]>, \Gamma_2 \to \Gamma_3}{\Gamma_1,<pdz_1...z_k,A>,\Gamma_2 \to \Gamma_3} \qquad \text{(D: left)}$$

$$\frac{\Gamma_1 \to \Gamma_2,<pdz_1...z_k,A>,<pd'u_1...u_k,A[d'/d,u_1/z_1,...,u_k/z_k]>,\Gamma_3}{\Gamma_1 \to \Gamma_2,<pdz_1...z_k,A>,\Gamma_3} \qquad \text{(D: right)}$$

Here p is a sequence (maybe, empty) of parameters and dummies, the most right variable of which (in non-empty case) is a parameter, d is a dummy, for i=1,...,k z_i is a dummy or parameter, and u_i is a new dummy or a parameter (in accordance with z_i).

In calculus mG' a deduction process starts with an input sequent of the form: $<p_1,M_1>,..., <p_m,M_m> \to <q_1,N_1>,...,<q_n,N_n>$, where $M_1 ,.., M_m, N_1 ,..., N_n$ are formulas without quantifiers, and $p_1,...,p_m,q_1,...,q_n$ are sequences of parameters and dummies, which are determined by the formula $(A_1 \wedge ... \wedge A_m) \supset (B_1 \vee ... \vee B_n)$, tested for validability, by the following way:

Let A'_1 , ... , A'_m, B'_1, ..., B'_n be some prefix normal forms of the formulas A_1, ..., A_m, B_1, ..., B_n, respectively. Then for every i=1,...,m (j=1,...,n) M_i is a matrix of A'_i (N_j is a matrix of B'_j), and p_i (q_j) is obtained by means of replacing in prefix of A'_i (B'_j) of every universal (existential) quantifier by a new dummy and of every existential (universal) quantifier by a new parameter.

All other notions (addmissible substitutions, deduction trees, proof trees, and so on) are the same as in the case of the calculus mG.

Corollary 2. For any formulas A_1, ..., A_m, B_1, ..., B_n the formula $(A_1 \wedge ... \wedge A_m) \supset (B_1 \vee ... \vee B_n)$ is valid if and only if there exists a proof tree for the input sequent $<p_1,M_1>,...,<p_m,M_m> \rightarrow <q_1,N_1>,...,<q_n,N_n>$ in the calculus mG'.

Proof.

The formula $(A_1 \wedge ... \wedge A_m) \supset (B_1 \vee ... \vee B_n)$ is valid if and only if $(A'_1 \wedge ... \wedge A'_m) \supset (B'_1 \vee ... \vee B'_n)$ is valid, where A'_1 , ... , A'_m, B'_1, ..., B'_n are prefix normal forms of A_1, ..., A_m, B_1, ..., B_n, respectively. It is easy to see that a proof tree for the input sequent $<p_1,M_1>,...,<p_m,M_m> \rightarrow <q_1,N_1>,...,<q_n,N_n>$ in mG' may be constructed on the basis of a proof tree for the input sequent $<,A'_1>$, ... , $<,A'_m> \rightarrow <,B'_1>,...,<,B'_n>$ and vice versa. To complete the proof, use Corollary 1. *Q.E.D.*

Remark. In calculus mG', the quantifier structures of formulas A_1, ..., A_m, B_1, ..., B_n are taken into account by means of sequences p_1, ..., p_m, q_1, ..., q_n. Selection of sequences for determination of quantifier dependencies does not play a principal role and was made for the purpose of visualizing and simplifying of the subject matter. It is possible to construct a (correct and complete) version of calculus mG' using analogs of "schemes" [5] instead of sequences (which also consist of parameters and dummies and reflect the quantifier structures of initial formulas more exactly) and modifying the rules (D: left) and (D: right). Observe also that Herbrand theorem in the form A from [5] may be easily obtained on the basis of a correctness and completeness of the version of calculus mG'.

Conclusion

In this paper the questions of implementation of computer-oriented sequent calculi are not considered because the development of efficient calculi requires optimizing the order of the propositional rule applications and selecting a method for generating of terms which may produce a proof tree. Bypassing details observe that for this purpose the unification algorithm combined with the introduced notion of admissible substitution is suitable. It was the approach that investigated at the level of modern vision [7] of the Evidence Algorithm programme, EA, advances by V. Glushkov. By now, the first version of the System for Automated Deduction, SAD, has been implemented (see Web-site ‘http://ea.unicyb.kiev.ua’). This implementation is based on a number of papers devoted to EA and SAD (see, for example, [8-10]).

Bibliography

1. Gentzen G. Untersuchungen uber das Logische Schliessen. Math. Zeit., 39, 1934, 176-210.
2. Gallier J.H. Logic for Computer Science: Foundations of Automatic Theorem Proving. - New York: Harper and Row, Inc., 1986, 513 pp.
3. Kanger S. Simplified Proof Method for Elementary Logic. - Comp. Program. and Form. Sys.: Stud. in Lodic, Amsterdam: North-Holland, Publ. Co., 1963, p. 87-93.
4. Lyaletski A.V. Variant of Herbrand Theorem for Formulas in Prefix Form (in Russian). – Kibernetika, 1981, No 1, p. 112-116.
5. Robinson J. A Machine-Oriented Logic Based on Resolution Principle. - J. of the ACM, 1965, p. 23-41.
6. Herbrand J. Recherches sur la Theorie de la Demonstration. - Travaux de la Societe des Sciences et de Lettres de Varsovie, Class III, Sciences Mathematiques et Physiques, 1930, 33.
7. Kapitonova Y., Letichevsky A., Lyaletski A., and Morokhovets. Algoritm Ochevidnosti - 2000 (project). - Proc. of the 1st Int. Conf. UkrPROG'98, Kiev, Ukraine, 1998, 68-70.
8. Degtyarev A., Lyaletski A., and Morokhovets M. Evidence Algorithm and Sequent Logical Inference Search. - Lecture Notes in Artificial Intelligence, 1705, 1999, 99-117.
9. Degtyarev A., Lyaletski A., and Morokhovets M. On the EA-Style Integrated Processing of Self-Contained Mathematical Texts.-Symbolic Computation and Automated Reasoning (the book devoted to the CALCULEMUS-2000 Symposium: edited by M. Kerber and M. Kohlhase), A K Peters, Ltd, USA, 2001, 126-141.
10. Verchinine K., Degtyarev A., Lyaletsky A., and Paskevich A. System for Automated Deduction (SAD): Linguistic and Deductive Peculiarities. - Advances in Soft Computing (Intelligent Information Systems 2002. (M.A.Klopotek, S.T.Wierzchon, M.Michalewicz (eds)). Physica/Springer Verlag, Heidelberg New York, ISBN 3-7908-1509-8, 2002, 413-422.

Author information

Alexander V. Lyaletski – Faculty of Cybernetics, Kiev Nashional Taras Shevchenko University, Senior Scientist Researcher; 2, Glushkov avenue, building 6, 03022 Kiev, Ukraine; e-mail: lav@unicyb.kiev.ua

THE SUBCLASSING ANOMALY IN COMPILER EVOLUTION

Atanas Radenski

Abstract. Subclassing in collections of related classes may require re-implementation of otherwise valid classes just because they utilize outdated parent classes, a phenomenon that is referred to as the subclassing anomaly. The subclassing anomaly is a serious problem since it can void the benefits of code reuse altogether. This paper offers an analysis of the subclassing anomaly in an evolving object-oriented compiler. The paper also outlines a solution for the subclassing anomaly that is based on alternative code reuse mechanism, named class overriding.

1 Introduction

Object-oriented applications are collections of related classes. For example, a typical compiler incorporates (1) a set of mutually recursive syntax trees and (2) translation operations on such trees; in an object-oriented compiler, such mutually related trees are implemented as mutually related classes.

As the requirements for an object-oriented application evolve, so should do the applications itself. For example, a programming language may need to be enhanced with new linguistic features, or it may need to have existing features modified. Consequently, an object-oriented compiler for such language may need to have some of its classes adequately adapted.

Subclassing is the principal object-oriented programming language feature that provides code adaptation. (Many patterns have evolved as more robust alternatives to straight forward subclassing for adaptation purposes, but in this paper we are interested in a discussion of linguistic primitives.) Subclassing allows the derivation of new classes from existing ones through extension and method overriding. A subclass can inherit variables and methods from a parent class, can extend the parent class with newly declared variables and methods, and can override inherited methods with newly declared ones.

When a class that needs to be updated belongs to a collection of classes but is independent from all other classes from the collection, the functionality of that class can be easily updated through subclassing and method overriding. Subclassing is a straightforward code adaptation mechanism in the case of independent classes.

Unfortunately, subclassing may not properly support code adaptation when there are dependencies between classes. More precisely, subclassing in collections of related classes may require re-implementation of otherwise valid classes just because they utilize outdated parent classes, a phenomenon that has been termed as the *subclassing anomaly* (Radenski 2002). The subclassing anomaly is a serious concern since it can largely invalidate the benefits of inheritance altogether.

The goal of this paper is to offer an analysis of the subclassing anomaly as it appears in an object-oriented compiler (Section 3). This analysis is preceded by an overview of the subclassing anomaly in domain-independent manner (Section 2). The paper outlines a solution to the subclassing anomaly based on an alternative code reuse mechanism called class overriding (Section 4), and concludes with a discussion of related work (Section 5).

2 Overview of the Subclassing Anomaly

Subclassing in a collection of dependent classes may require re-implementation of otherwise valid classes just because they depend on the parent class. The need to re-implement such otherwise valid classes is referred to as the subclassing anomaly. The subclassing anomaly needs to be understood because it may seriously affect code reusability. This section is devoted to a brief overview of the subclassing anomaly. A more detailed analysis of the subclassing anomaly in a problem independent manner is presented in (Radenski 2002).

Depending on the programming language, a collection of classes can be represented as a namespace (in C#), a stateless package (in Java), or as a package with a state (in Ada 95). In this paper we utilize C# as sample language in order to provide clarity of discussion. However, all results presented in the paper can be applied equally well to virtually any compiled object-oriented language.

Let us assume that in a collection of related classes, a *container* class instantiates and utilizes an object of a *constituent* class. Let us also assume that at a later point of the existence of the collection of classes, the constituent class needs to be adapted to changing requirements, while the container class remains valid, meaning that it still provides relevant functionality and needs no changes.

Subclassing of the constituent produces an evolved constituent subclass of the original constituent class, which is then incorporated in the evolved collection. The problem is that the integrity of the evolved collection is violated, since in the evolved collection the container class still instantiates and utilizes an object of the old parent constituent class, rather than an object of the evolved constituent class. Even though the container class is assumed to provide relevant functionality, it needs to be re-implemented (which is anomaly), so that it creates an object of the evolved constituent class and thus maintains the integrity of the evolved collection.

Classes may depend on each other in various ways. Some dependencies do not cause anomalies, while others do. The so-called monomorphic dependencies, as defined below, trigger the subclassing anomaly.

Object-oriented languages allow two types of references to classes: polymorphic references and monomorphic references. A *polymorphic reference* to a class *C* stands (1) for *C* itself and (2) for all possible subclasses of *C*. A *monomorphic reference* to a class *C* stands for *C* only but not for any subclasses of *C*.

Polymorphic references to a class *C* occur in:

- parameter, variable, and constant declarations, e.g.: void f (C x); C x;
- type tests, e.g.: if (y is C) …; if (y instanceof C) …;
- type casts, e.g.: x = (C) y;

Monomorphic references to a class *C* occur in:

- constructor invocations, e.g.: *x* = *new C ();*
- static member access, e.g.: *C.staticMethod ();*
- subclass definitions, e.g.: *class C1 : C {…} ; class C1 extends C {…};*

A class *A* depends monomorphically on class *C* if the definition of *A* contains a monomorphic reference to *C*; further on, we skip the word monomorphically and simply say that *A depends on C*. A class *A* depends on *C* when *A* invokes the constructor of *C*, when *A* extends *C*, or when *A* refers to a static member of *C*.

The subclassing anomaly is triggered by monomorphic dependencies within a collection of classes. When the collection evolves, subclasses can be defined in order to adapt the collection to the changing environment. However, no matter how subclassing is applied, a monomorphic reference continues to stand for the outdated base class in the evolved collection. Thus, all classes that contain monomorphic references must be re-implemented, often in textually equivalent form, as members of the evolved collection. Such re-implemented classes must be recompiled so that monomorphic references are bound to up-to-date subclasses. In contrast to monomorphic references, polymorphic references to outdated base classes do not necessarily require re-implementation of the referring classes - because polymorphic references stand not only for the base class (as monomorphic references do), but for all of its subclasses as well.

3 Analysis of the Subclassing Anomaly in an Evolving Object-Oriented Compiler

This section is devoted to an analysis of the subclassing anomaly in an evolving object-oriented compiler. Our goal is to provide a non-trivial example of the subclassing anomaly as defined in the previous section and to reveal various kinds of class references that trigger the anomaly.

This is not an artificially constructed design example: it is derived from a popular book on object-oriented compilers (Watt, 2000). Depending on one’s personal perspective, this design might be considered bad or good (we consider it good), but what is more important, is that it is *common* design which exhibits the subclassing anomaly.

The sample compiler has the usual three phases of syntactic analysis, contextual analysis, and code generation. As shown on Fig. 1, the three phases are implemented as *Parser*, *Checker*, and *Encoder* objects. The parser, checker, and encoder take one pass each, communicating via a syntax tree that represents the source program.

Syntax trees are defined as a hierarchical collection of interfaces and classes. On the top of the hierarchy, a *SyntaxTree* interface encapsulates methods common for all abstract syntax trees (such as a visitor method implemented by both the contextual analyzer and the code generator). Any multiple-form non-terminal symbol is represented by a single interface and several classes that implement this interface, one for each form. For example, statements are represented by the *Statement* interface and several classes that implement this interface, such as *WhileStatement*, *IfStatement*, etc.

The recursive-descent *Parser* class consists of a group of methods *parseN*, one for each non-terminal symbol *N*. The task of each *parseN* method is to performs syntactical analysis of a single *N*-form, and build and return its syntax tree. These parsing methods cooperate to perform syntactical analysis of a complete program. For example, *parseWhileStatement* performs syntactical analysis of a single *WhileStatement,* and creates and returns an instance of a *WhileStatement* syntax tree (Fig. 1).

```
namespace CompilerCollection {
      public class Compiler {
                        public static void compileProgram (...)
                        {      Parser parser = new Parser (...);
                               Checker checker = new Checker (...);
                               Encoder encoder = new Encoder (...);
                               Program syntaxTree = parser.parse (…);
                               checker.check (syntaxTree);
                               encoder.encode (syntaxTree);
                        }
      }
      public abstract class SyntaxTree {… }
      public abstract class Statement : SyntaxTree {…}
      public class WhileStatement : Statement
                  { Expression e; Statement s; …}
                  …
      public class Parser {…
                        public Statement parseWhileStatement ()
                        {      … Expression e = parseExpression ();
                               … Statement s = parseStatement ();
                               … return new WhileStatement (e, s);
                        }
                        ….
      }
      public class Checker {…}
      public class Encoder {…}
}
```

```
using CompilerCollection;
namespace UpdatedCompiler {
                  public           class           WhileStatement           :
CompilerCollection.WhileStatement
                  { public void display () {…} }
                  public class IfStatement : CompilerCollection.IfStatement
                  { public void display () {…} }
                  …
                  public class Parser {… // identically re-implemented
                        public Statement parseWhileStatement ()
                        {      … Expression e = parseExpression ();
                               … Statement s = parseStatement ();
                               … return new WhileStatement (e, s);
                        }
```

Anomaly triggered by constructor invocation. Suppose that a developer needs to enhance all syntax trees classes with a *display* method, thus converting the *CompilerCollection* into an *UpdatedCompiler* (Fig. 1). One approach is to use subclassing in order to extend with a *display* method all original syntax tree classes, such as *WhileStatement*, *IfStatement*, etc.

Unfortunately, subclassing of the syntax tree classes does not affect any other classes form the *CompilerCollection* and all *parseN* methods from the *Parser* class continue to instantiate the old syntax tree classes. For example, the *parseWhileStatement* method form the *Parser* class, as defined in the *CompilerCollection*, instantiates class *WhileStatement* which is also defined in the *CompilerCollection*.

To effectively update the *CompilerCollection* and convert it into an *UpdatedCompiler*, the developer needs to re-implement the otherwise valid *Parser* class. The re-implementation of the *Parser* class is textually identical with the old one and only needs to be encapsulated within the *UpdatedCompiler*.

The necessity to re-implement a valid *Parser* class is triggered by the subclassing of the syntax tree classes, and this phenomenon is an example of the subclassing anomaly. Note that each *parseN* method from the

Parser class instantiates an object of class *N*, where *N* is the syntax tree representation for *N*. These monomorphic dependencies of class *Parser* on classes *N* trigger the inheritance anomaly.

Anomaly triggered by subclass definition. A developer who needs to enhance all syntax trees classes with a *display* method needs to start with their parent class. Technically, the developer should use subclassing in the *UpdatedCompiler* in order to extend the *SyntaxTree* abstract class with a display method. Unfortunately, the *Statement* subclass of the original *SyntaxTree* class is not affected by this subclassing and remains without a *display* method, as originally defined in *CompilerCollection*. Therefore, the developer needs to re-implement in the *UpdatedCompiler* the otherwise valid *Statement* class. The re-implementation of the *Statement* class is textually identical with the old one and only needs to be encapsulated within the *UpdatedCompiler*.

The necessity to re-implement a valid *Statement* class is triggered by the subclassing of the *SyntaxTree* class, and this phenomenon is an example of the subclassing anomaly. Note that the *Statement* class is defined in the *CompilerCollection* as a subclass of *SyntaxTree*. This monomorphic dependency of the *Statement* class on the *SyntaxTree* class triggers the inheritance anomaly.

Anomaly triggered by static member access. A compiler may use classes with static members for various purposes. For example, all token kinds may be specified as static members of a *Token* class and all operation codes can be encapsulated as static members of a *Machine* class. Parser methods need to access static members of the *Token* class, e.g. *Token.While*. Suppose now that a developer of an *UpdatedCompiler* needs to enhance the *Token* class with a new token, such as *Repeat*. One approach is to use subclassing in order to extend *Token* with a *Repeat* static member. Unfortunately, subclassing of the *Token* class does not affect any of the static references to the original *Token* class, as defined in the *CompilerCollection*. All inherited parser methods continue to use the old version of the *Token class*, while new parser methods that are developed in the *UpdatedCompiler* utilize the updated *Token* class. If the developer wants to have the parser utilize the same *Token* class, the developer must re-implement the whole *Parser* class in the *UpdatedCompiler*. The re-implementations of all inherited parser methods are textually identical with the old ones and only need to be encapsulated within the updated *Parser* class.

The necessity to re-implement valid parser methods is triggered by the subclassing of the *Token* class, and this phenomenon is an example of the subclassing anomaly. Note that the parser methods access static members of *the Token* class. This monomorphic dependency of the *Parser* class on the *Token* class triggers the subclassing anomaly.

4 Elimination of the Subclassing Anomaly with Class Overriding

Class overriding, an object-oriented language feature that is complementary to subclassing, can be used to eliminate the subclassing anomaly (Radenski 2002). In contrast to subclassing, class overriding does not create a new and isolated derived class, but rather extends and updates an existing class. Class overriding is not limited to a single class but propagates across a collection of related classes: it updates all classes from the collection that refer to the class being overridden. Thus, class overriding preserves the integrity of a collection of classes by guaranteeing that any update to a class replaces the previous version of the class within the whole collection.

The definition of class overriding is based on the concept of *replication*. Replication consists in embedding a replica of each class from an existing collection of classes (*the replicated collection*) into a newly created collection of classes (*the replicating collection*). In addition to class replicas, the replicating collection can be further extended with newly defined classed or subclasses.

Replication changes class membership: while all original classes are members of the replicated collection, the class replicas become members of the replicating collection. Except for class membership, class replication preserves all other class properties, including names and access levels. In the replicating collection, each class replica is referred to by the same name and incorporates the same public, protected, and private access levels as the original class in the replicated collection.

A class replica can be overridden (meaning replaced) across the entire replicated collection with its own extension. Similarly to a subclass, the overriding class:

- inherits all data and method members of the class replica
- can override some of the inherited methods
- can extend the replica with additional data and method members

The overriding class replaces the class replica across the entire replicated collection, meaning that all classes from the replicated collection are updated to use the overriding class instead of the replica. Technically this is achieved by *late class binding*: class references are bound to particular class definitions late, at class loading time, rather than early, at compile time. This is in contrast to traditional compiled languages, such as C#, which use late binding only for methods but limits monomorphic class references to early static binding.

C#, and likewise, various other object-oriented languages, can be enhanced to support class overriding. In C#, collections of classes can be represented as namespaces. Therefore, C# is to be extended with namespace replication statements and with class overriding definitions.

```
namespace CompilerCollection {
       public class Compiler { … }
       public abstract class SyntaxTree {… }
       public abstract class Statement : SyntaxTree {…}
       public class WhileStatement : Statement { Expression e; Statement s; …}
                     …
       public class Parser {… }
       public class Checker {…}
       public class Encoder {…}
}
```

```
namespace UpdatedCompiler {
                     replicate CompilerCollection;
                     override public class WhileStatement  { public void display
() {…} }
                     override public class IfStatement { public void display ()
{…} }
                     …
}
```

Figure 2. Elimination of the subclassing anomaly by namespace replication and class overriding.

A C# outline of a compiler that is updated by means of namespace replication and class overriding – thus avoiding the subclassing anomaly - is presented in Fig. 2. Class overriding updates the *WhileStatement* and *IfStatement* across the entire replicated *CompilerCollection.* No re-implementation of valid classes is needed.

5 Conclusions

This extensibility problem (Findler, 1999; Flatt 1999) appears when a recursively defined set of data and related operations are to be extended with new data variants or new operations. A set of recursive data and related operations can be straightforwardly represented as a collection of dependent classes. Thus, compiler extensibility can be viewed as a special case of recursive class extensibility. Although extensibility can be achieved through subclassing, it requires extensive use of type casts and cumbersome adaptation code, a necessity that is referred to as the extensibility problem.

The compiler extensibility problem can be avoided by following design patterns that are targeted specially at extensibility, such as the extensible visitor (Krishnamurthi et al., 1998), the generic visitors (Palsberg and Jay, 1997), and the translator pattern (Kühne, 1997). Using such patterns implies serious penalties. In the case of the extensible visitor and the translator patterns, the penalty is the significant programming effort needed for an extension. In the case of the generic visitors, the penalty is the significant run-time overhead imposed by the utilization of reflectivity.

Several known linguistic techniques can be applied to attack the compiler extensibility problem, as for example the extensible data types with defaults of Zenger and Odersky (2001) and the evolving open classes of Clifton et al. (2000). None of the known language-level mechanisms seems to offers a silver bullet solution for software evolution. Compared to other approaches, class overriding is simpler and easier to use method to eliminate the subclassing anomaly.

Acknowledgement

This research has been partially supported by a National Science Foundation grant, award number 0243284.

References

1. Clifton, C., G. Leavens, C. Chambers, T. Millstein, 2000. MultiJava: Modular Open Classes and Symmetric Multiple Dispatch for Java. OOPSLA'00, Minneapolis, Minnesota, October 2000, ACM Press, New York, 130-145. http://www.cs.iastate.edu/~cclifton/multijava/papers/TR00-06.pdf
2. Findler, R., M. Flatt, 1999. Modular Object-Oriented Programming with Units and Mixins. ACM SIGPLAN International Conference on Functional Programming (ICFP '98), 34(1), 94-104. http://www.cs.utah.edu/plt/publications/icfp98-ff/icfp98-ff.pdf
3. Flatt, M., 1999. Programming Languages for Reusable Software Components. PhD thesis, Rice University, Houston, Texas.
http://cs-tr.cs.rice.edu/Dienst/UI/2.0/Describe/ncstrl.rice_cs/TR99-345/
4. Krishnamurthi, S., M. Felleisen, D. P. Friedman, 1998. Synthesizing Object-Oriented and Functional Design to Promote Reuse. ECOOP'98, Brussels, Belgium, July 1998, Springer, Berlin, 91-113.
5. Kühne, T., 1997. The Translator Pattern - External Functionality with Homomorphic Mappings. In Ege, R., M. Singh, and B. Meyer (Eds.), The 23rd TOOLS conference USA 1997, 48-62.
6. Palsberg, J., C. B. Jay, 1997. The Essence of the Visitor Pattern. Technical Report 05, University of Technology, Sydney, Australia.
7. Radenski, A., 2002. Anomaly-Free Component Adaptation with Class Overriding, Journal of Systems and Software, Elsevier Science (under print).
8. Zenger, M., M. Odersky, 2001. Extensible Algebraic Datatypes with Defaults. International Conference on Functional Programming, ICFP 2001, Firenze, Italy, September, 2001. http://lamp.epfl.ch/~zenger/papers/icfp01.pdf
9. Watt, D., D. Brown, 2000. Programming Language Processors in Java: Compilers and Interpreters, Prentice Hall, New York, New York.

Author Information

Atanas Radenski - Chapman University, One University Drive, Orange, California 92866, USA
radenski@computer.org; http://www.chapman.edu/~radenski/

FRONTAL SOLUTIONS: AN INFORMATION TECHNOLOGY TRANSFER TO ABSTRACT MATHEMATICS

V. Jotsov

Abstract: The paper introduces a method for dependencies discovery during human-machine interaction. It is based on an analysis of numerical data sets in knowledge-poor environments. The driven procedures are independent and they interact on a competitive principle. The research focuses on seven of them. The application is in Number Theory.

Keywords: knowledge discovery and data mining, modeling, Number Theory.

1. Introduction

The offered research has begun since 1986 after the exploration of some of the early D. Lenat's papers [Lenat 1976, Lenat 1983]. They gave us the conviction, that the information technologies (IT) are suitable for applications in models which are bounded by Number Theory. The newest evolutionary programming (EP) [EAEA 1997, EA 1997, Nordin 1999] research confirms the possibilities for elaborating new formulas. The considered paper follows the line from our papers [Jotsov1 1999, Jotsov2 1999]. Compared with the works of Lenat [Lenat 1983], or with other sources in the references on informatics, the majority of our papers describe the mathematical results, not the method. The paper's scope is *interdisciplinary* and includes many significantly far research areas. To some extent the proposed method is a continuation of the Lenat's ideas and serves the same **purposes**: elicitation of new knowledge in the integer data processing, derivation of new formulas, and *whenever possible* generation of new mathematical theorems. At the same time it has some points in common with the Narin'yani's, Shvetsov's constraint programming [Narin'yani 2000,Shvetsov 1997] and reasoning in the Altshuller or Hadamard or Polya style [Altshuller 1979, Hadamard 1975, Polya 1963]. The approach is enriched from the most remote principles coming from both directions but it *uses no plausible reasoning*.

2. The FRONTAL Method and the Working Environment

The shortly described below FRONTAL method interacts with several other methods under the common control of a new type of an evolutionary metamethod. The metamethod avoids or *defeats* crossovers, phenotypes, mutations, etc. Below we choose the description in an analogous manner as the way to reduce the extra descriptions, because the general scheme of the chosen strategy is rather voluminous. The evolutionary metamethod swallows and controls the following methods:

I. FRONTAL method;
II. KALEIDOSCOPE method;
III. FUNNEL method;
IV. CROSSWORD method.

The KALEIDOSCOPE method is the *background* for the human-machine strategies for work. The machine forms and visualizes different mappings for the chosen groups of numbers or like, while the obtained results are estimated by the human. *The human* makes the necessary conclusions and undertakes the required steps. Analogically the kaleidoscope rotations form different images in a hazardous manner, and the spectator takes an *informal decision* whether the seen by him is nice, original etc.

Let's assume you have a *plastic funnel*. If you fix it vertically above the ground, you can direct a stream of water or of vaporous drops etc. If you change the funnel direction, then the stream targeting will be hampered. Fixing the funnel horizontally makes it practically useless. Analogically in the evolutionary method the general direction in numerical models is determined likewise. In other words this is a movement along the predefined gradient of the information. This term is proposed in a manner which *has some connection* to [Baldi 1995]. Just like in the case of the physical example in the beginning of the investigation there are lots of undirected hazardous steps towards conclusions and hypotheses. The FUNNEL method is based on inconsistency tests with known information.

Let us assume that the reader solves a problem with a complex sentence of 400 letters with vague for the reader explanations. Let the unknown sentence be horizontally located. The reader can't solve the problem in an arbitrary manner, because the number of combinations is increased exponentially. Now it is convenient to **facilitate** the solution by linking the well known to the reader information with the complex one from the same model. The reader tries to find vertical words that he is conscious about like the place of our conference KDS 2003 - Varna. The more the crosspoints are, the easier is the solution of the horizontal sentence. The approach for the CROSSWORD is *even easier*. Here both the easy meanings and the difficult ones are from one domain, therefore there exists an additional help to find the final solution.

For pity the paper length does not allow us to make more detailed descriptions of the mentioned above methods, and/or their connections, interactions, etc. We will turn exclusively to the considered FRONTAL method.

The trend in the investigation includes solutions of complex hypotheses and problems which require the usage of integer-number models. Great number of these problems have been unsolved for centuries; their decisions cannot be obtained *prima vista* or in a **frontal** manner. This is the reason for the development and application in mathematics of an evolutionary strategy. In it the **preproofs** are on the first place. In the process of solving oversophisticated problems the first draft solutions comprise only the first step in the marked by the FUNNEL direction. This direction is an approximate. This is due to the initial conditions and knowledge constraints. Fig. 1 depicts a similar general direction for research by the A-B line. The obtained intermediate solutions follow another route, A-C-D-B. The solution B is inaccessible from the node C or from any other node before D. The user can change the direction according to her/his wish. D-E on Fig. 1 is a deviation from the line A-B. The new branch marks the process of solving another problem. Any of the intermediate solutions may contradict or doesn't correspond to the final solution (B). Together they form the set of preproofs for B. The mathematical proofs are formed in the process of evolution with no probabilities. In the evolutionary metamethod the preproofs are usually weak, with bottlenecks and/or incomplete. The preproofs in the considered domain are never so good as to be included in the "official" proof. Nevertheless they must not be easily rejected. They are weaker, but in our case they *are not* heuristical by nature, and they might assist the solution of other problems as well.

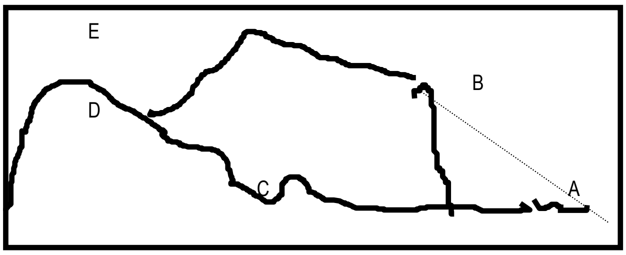


Fig. 1.

The presented evolutionary meta-method has the following features. The solution is evolved *step by step*. At every step it is possible to have a progress or a regress compared with the previous decision. The role of probabilities and other subjective estimations is played by interactive approaches for knowledge acquisition, data linkage, mappings and other processing of data and knowledge. The investigated FRONTAL method (I) includes the following procedures. Their short abbreviations are given in bold letters.

1. **MOC**: Mix Or Change (data/knowledge);
2. **BIND**: Connects the information (data sets/knowledge) during the automatic work or shows it to the user;
3. **WHY & HOW**: Forwards it (data sets/knowledge) to the user;
4. **CS**: Constraint Satisfaction (of knowledge), based on the weak negation ~;
5. **SPREAD** (knowledge);
6. **WHAT**: Explanation (of data/knowledge);
7. **EF**: Elimination Filter.

All the seven procedures can be modified together with the change of the different models. Now we introduce in short the FRONTAL method terminology. Let M be a set of such models M_i which contain sets of arithmetic progressions $\{a_i+b_ik\}^{\infty}_{k=0}$. At that:

(1) $$b_i = \prod_{p_{j\in M}} p_j; \quad p_j \in P.$$

where P denotes the prime numbers set. Every progression from M_i may be treated as a result after sieving out the set of positive integers, consisting of all p^i_M and such composite numbers that at least one of p^i_M divides them. To simplify the contents other models are not included, e.g. based on geometrical progressions. It is accepted that $(a_i,b_i)=1$; $a_i<b_i$.

Four operations are introduced in every model: {+,-,*,/}. Possibly every application of the algorithms based on the FRONTAL method leads to some change of different parameters inside the built-up algorithms whenever the model changes. This model changes serve as *an algorithm stability test*. This is the right place to use MOC. Denote $V=\{v_1,v_2,...v_z\}$ is a set of parameters. During our first investigations in the eighties we used V in a way similar to the genotype from Genetic Algorithms (GA). The user had the option to accept such v_i which deserved his attention and the system proceeded with the goal task. We offered that every task must begin with $V=\{\varnothing\}$. Thus the *released assumption* brings the user closer to data mining tasks.

The author proposes the following generalized MOC algorithm with an automatic mode set-up: **A**. Fixing of v_i in the current model; **B**. Case-based inclusion of v_i from previous solutions; **C**. The algorithm proceeds with review of v_i =0; **D**. An inverse mapping of (C.) is introduced or $v_i \rightarrow$max; **E**. v_i is replaced by another parameter in V; **F**. The algorithm goes on with the WHAT procedure or with other procedures from the FORWARD method. The general MOC scheme is postulated with the formulas (2) and (3).

(2) $$S(V\rightarrow V'); \ card(V)\neq card(V').$$

(3) $$L(S(v_{i,k}))\rightarrow L(S(v_{i,j})); \ S(v_{i,k})\neq S(v_{i,j}).$$

Here S is a situation which has arisen as a result from the MOC activity changing the set V or its separate element v_i. L is the modal operator *possibility*.

For example, let v_5=2 means that all the numerical data are copied in a bidimensional array. This automatically inputs $v_6 = \overrightarrow{x}$ and $v_7= \overrightarrow{y}$ in V. During the activation of (**D**.) $v_8= \overrightarrow{z}$ is introduced, etc. When processing (**C**.), the bounded with v_5 parameters v_6=0 or v_7=0 are affected. In this way MOC acquires new knowledge from the data investigation. The next example is not so theoretical. Rather it is connected with numbers from eight arithmetic progressions.

The following denotations are introduced. $\mathbf{\{}m+nk\mathbf{\}}^{\infty}_{\mathbf{k=0}}$ is an arithmetic progression (progression for short). In it m is the first member, and n is the step. $\boldsymbol{\pi(x)}$ is the total number of the primes which are elements of the set **P** ($p_i \in P$, $p_i \leq x$). $\boldsymbol{\pi_{n,m}(x)}$ is the number of primes $\leq$x which are contained in the progression. $\mathbf{S_5}$ is an union of 8 progressions $\{y+nk\}^{\infty}_{k=0}$, $y \in Y$, Y={1,7,11,13,17,19,23,29}. Every of these progressions is represented as a column in Fig. 2 if the elements of S_5 are shown vertically. Fig. 3 shows the same environment in a slightly different manner. Every of the elements in S_5 is computed in the following way. The first number from the corresponding column - see line 1 - is added to the number from the same line and the leftmost column. For example $s_{14,2}$=7+390 is in line 14 and column 2. Composite numbers in S_5 are represented as products of prime numbers. The primes are the result of the decomposition of the composites. In Fig. 3 the primes are *omitted* while the particular cases $y \in Y$ are given in brackets. MOC has no logical inference. It simply finds and changes the scope parameters one by one while the rest of the parameters remain unchanged. The lines below show the cases when MOC pastes or cuts some of the elements in the interpretation. For example during the investigation of the operations addition and multiplication in S_5 the following parameters attract the attention: primes (with just a single divisor), composites with at least 2 divisors, 8 columns which are parallel to the vertical axis $\overrightarrow{y}$ and 15 lines which are parallel to $\overrightarrow{x}$. These 4 parameters can have other designations, which will have similar meanings. The names are not significant. The parameters are established by mere observations e.g. directly on the figures. The following transforms for the transition from Fig. 2 to Fig. 3 are used:

(T_1). The primes are determined but not shown from all the numbers in the fragment, see Fig. 2. The *very omission* introduces some new information. The figures below demonstrate the following versions of transformations in S_5.

(T_2). All the composites are presented as products of prime divisors.

(T_3). All the composites with the divisor of 13 are successively connected with straight lines.

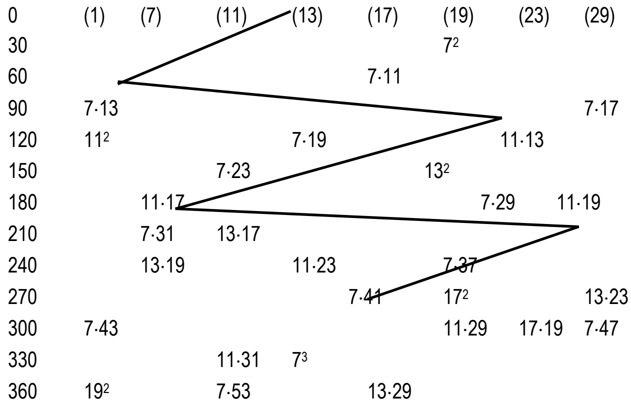
1	7	11	13	17	19	23	29
31	37	41	43	47	49	53	59
61	67	71	73	77	79	83	89
91	97	101	103	107	109	113	119
121	127	131	133	137	139	143	149
151	157	161	163	167	169	173	179
181	187	191	193	197	199	203	209
211	217	219	223	227	229	233	239
241	247	251	253	257	259	263	269
271	277	281	283	287	289	293	299
301	307	311	313	317	319	323	329
331	337	341	343	347	349	353	359
361	367	371	373	377	379	383	389
391	397	401	403	407	409	413	419
421	427	431	433	437	439	443	449

Fig. 2.

(T_4). All the composites with the divisor of 17 are successively connected with straight lines. The result is shown in Fig. 4. The transformation itself is in the divisor replacement.

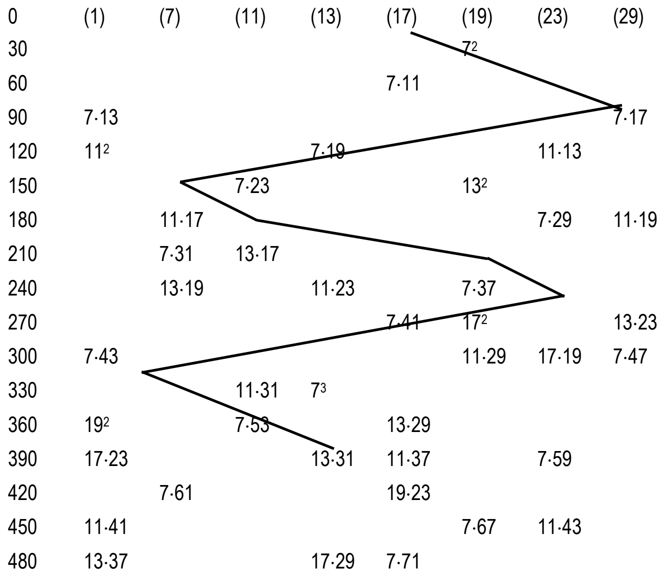
(T_5). Besides the graphical interpretations in Fig 3 and Fig. 4 must be added similar pictures for the “neighbors below” 43 and 47 or 13+30, 17+30. The result has the same succession of beat for the columns with periods 30 times 43 and 30 times 47. The illustrations resemble the Fig. 3 and Fig. 4 but they are more elongated due to the greater period.

(T_6). The parameter influence of $\overrightarrow{x}$ is **"reduced".** So the attention is concentrated upon the **beat succession** for the columns S_5 and the lines are "compressed". The results are depicted in Fig. 5 and Fig. 6.



0	(1)	(7)	(11)	(13)	(17)	(19)	(23)	(29)
30						7^2		
60					$7\cdot 11$			
90	$7\cdot 13$							$7\cdot 17$
120	11^2			$7\cdot 19$			$11\cdot 13$	
150			$7\cdot 23$			13^2		
180		$11\cdot 17$					$7\cdot 29$	$11\cdot 19$
210		$7\cdot 31$	$13\cdot 17$					
240		$13\cdot 19$		$11\cdot 23$		$7\cdot 37$		
270					$7\cdot 41$	17^2		$13\cdot 23$
300	$7\cdot 43$					$11\cdot 29$	$17\cdot 19$	$7\cdot 47$
330			$11\cdot 31$	7^3				
360	19^2		$7\cdot 53$		$13\cdot 29$			

Fig. 3.



0	(1)	(7)	(11)	(13)	(17)	(19)	(23)	(29)
30						7^2		
60					$7\cdot 11$			
90	$7\cdot 13$							$7\cdot 17$
120	11^2			$7\cdot 19$			$11\cdot 13$	
150			$7\cdot 23$			13^2		
180		$11\cdot 17$					$7\cdot 29$	$11\cdot 19$
210		$7\cdot 31$	$13\cdot 17$					
240		$13\cdot 19$		$11\cdot 23$		$7\cdot 37$		
270					$7\cdot 41$	17^2		$13\cdot 23$
300	$7\cdot 43$					$11\cdot 29$	$17\cdot 19$	$7\cdot 47$
330			$11\cdot 31$	7^3				
360	19^2		$7\cdot 53$		$13\cdot 29$			
390	$17\cdot 23$			$13\cdot 31$	$11\cdot 37$		$7\cdot 59$	
420		$7\cdot 61$			$19\cdot 23$			
450	$11\cdot 41$					$7\cdot 67$	$11\cdot 43$	
480	$13\cdot 37$			$17\cdot 29$	$7\cdot 71$			

Fig. 4.

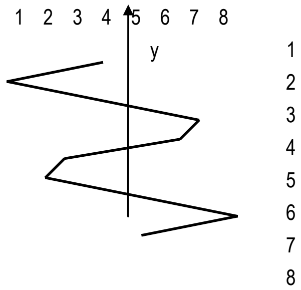


Fig. 5.

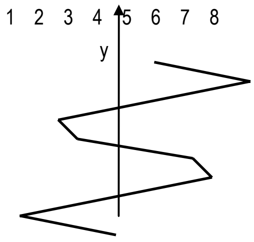


Fig. 6.

The discussed six relatively simple transformations show plainly and unambiguously that the cited in Fig. 5 way to beat the columns is one and the same for all the elements in column 4 in S_5: 13, 43... The result is in relation with the transition from a piece of S_5 to the whole S_5 or v.v. It is specially discussed in the SPREAD presentation. Fig. 6 presents the situation with the elements in column 5 (17,47...) which is analogous.

The two numerical sets in Fig. 5 and in Fig. 6 interpret the same cycles as those in Fig. 3 and Fig. 4. These cycles have common “similarity centers” on $\vec{y}$. Moreover the two figures coincide if one of them is rotated 180 degrees around $\vec{y}$ (T_7). The revealed dependency is valid only for numbers of the type n and 30k-n for every positive integer k. If the beat cycle for the columns in Fig. 5 is in a column starting with the element m, then the analogical cycle in Fig. 6 is in a column starting with 30-m. The constantly repeated number 30 leads to (T_8): $30=2\cdot3\cdot5$. *The act of mathematical creation* for Fig. 2-Fig. 3 is unambiguously simple when mapping Fig. 5 to Fig. 6. The revealing of different numerical properties takes place in the described above MOC procedure. Other transformations can be pointed like (T_9): the discovery of numbers which *can’t be divisors* of any integer number. Zero which is not an element in S_5, but being a similarity center for the positive and negative parts in S_5, is set in this manner. The interpretation of any prime cycle as on Fig. 7 is unified by the total discrimination of the influence of $\vec{y}$; (T_{10}) is a suitable example as an illustration vs. (T_7).

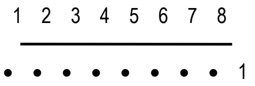


Fig. 7.

All discussed transformations are just consequences of observations based on the model. They give no answers to questions like WHY or HOW the presented results are obtained. The body of the preproofs is formed on the basis of such conclusions.

The achieved with MOC results may be related and compared. This is the purpose of the BIND procedure. The extracted information is analyzed by BIND on the basis of juxtapositions. BIND is based on the above function mapping $S_{x1}(v_1,...v_z)=S_{x2}(v_1,...v_z)$ or $S_{x1}(v_1,...v_z)\neq S_{x2}(v_1,...v_z)$ where x_i are different objects or data groups. The detailed BIND overview exceeds consideration line in the paper. The obtained results most of all lack of proving power and the inference obtained is nonmonotonous. Therefore after determining the regularities it is possible to formulate prompting queries to the user which are decorated in the well known form **WHY** and **HOW.** The system forms the basis for the general solution, and the details are an object for a manual or an interactive work. In this way, the investigation evolves itself. Using the WHY&HOW procedure, a new set is built from mutually related formulas and knowledge from the same domain.

The CS procedure is formalized in a manner similar to the one in [Narin’yani 2000]. An outstanding feature of the presented variant of CS is that the bounds of the domain are not restrictive in the case of a weak negation ~. After the contradictory resolution these bounds are overcome. The contradiction concentrates the attention to the incompleteness in the scope. The goal-forming scenario in the constraint satisfaction paradigm is formulated as follows. Let the variables x_1, x_2, ... x_n be the mapped sets of their value spaces X_1, X_2, ... X_n. The constraints $C_j(x_1,x_2,...,x_n)$, j=1,...k are valid for the same X_j. It is necessary to find such sets $<a_1, a_2, ... a_n>$ such that $a_i\in X_i$ and they satisfy all C_j simultaneously.

Denote M* is a *subdefinite model* or - roughly speaking - an incomplete model. Let $C_i(x_1,x_2,...,x_n)$ is one from the investigated constraints, and $N(x_1,x_2,...,x_n)$ be such that:

$$(4)\qquad N(x_1,x_2,...,x_n) \rightarrow \sim C_i(x_1,x_2,...,x_n).$$

This means that the constraint is violated because (4) contains the weak nonclassical negation ~. The ~ based inconsistencies may be solved after the complementation of M* with new knowledge/data. The augmented model is denoted with M’. In it the examined constraint takes the form $C'_i(x_1,x_2,...,x_n)$, z=n or z≠n, where:

$$(5)\qquad C'_i(x_1,x_2,...,x_n) \rightarrow \sim C_i(x_1,x_2,...,x_n).$$

There are other possible ways for the transition M*-M' besides $N(x_1,x_2,...,x_n)$. One of them is to include a new parameter v' in M*. Another approach is possible in the case when the system of constraints has no solution. Often in such cases there exists an information which admits the re-examination of $C_i(x_1,x_2,...,x_n)$. For example, let us examine the numbers x≥11. Then we may come to the conclusion that:

(6) $$\pi(x) > \frac{x}{\ln(x)}.$$

Here M* has no constraints and $C_i=\{\varnothing\}$. The result can be monotonously generalized to the whole interval [0,∞]. The case when x=8 violates the formula (6). This contradicts the assumptions especially the case $C_i=\{\varnothing\}$. The introduction of C'_1: x≥11 leads to the result:

(7) $$\pi(x) \underset{x\geq 11}{>} \frac{x}{\ln(x)}.$$

The last three procedures do not contain substantially new theoretical ideas. SPREAD is based on the well known concept of mathematical induction. WHAT is designed to communicate with humans, because the internal representation of the solutions is obscure. EP serves as a barrier against knowledge duplications or a surplus knowledge.

The interaction between the first five procedures is on a competitive basis according to the JUNGLE principle. In some cases they act in the role of *demons*. In the rest of the cases the top priority is assigned to the procedure from the previous iteration or this one which has generated the most effective solutions. The following formalization is aimed to derive this simple estimates and agreements. JUNGLE is based upon estimates $0\leq f(Q_i)\leq 1$ for every procedure of the FRONTAL-based set $Q=\{Q_1,...,Q_7\}$. In this case it is preferable to compare the described JUNGLE strategy with the one from GA "the fittest wins" ([EAEA 1997], p.3). We use it in the form "the winner is best estimated". If $f(Q_i)=1$, then the procedure interacts with EF and the user. If $0.25\leq f(Q_i)<1$, then the display contains this value, and the corresponding solutions are considered only on the user request. The user may interfere in the automatic process of the estimation. The threshold value $f(Q_i)=1$ is achieved in the following situations:

(8) $S(Q_j)\rightarrow G(Q_i)$; $j\neq i$; $i,j=1...7$; $G(Q_i)\rightarrow f(Q_i)=1$.

where G is the modal operator *necessity*, $S(Q_j)$ is a scenario in Q_j leading to $G(Q_i)$. An example of (8) is presented above after (**T_6**) thus activating SPREAD by MOC.

(9) $U\rightarrow f(Q_i)=1$.

Here U means *user-defined activation*. The user defines the necessary parameters for Q_i.

(10) $S(Q_3)=c \rightarrow f(Q_i)=1$.

where $S(Q_3)$ is the BIND output. The meaning of **c** (for short from **convergence**) is that *the results from the two independent research lines coincide*. Fig. 7 depicts an example leading to $S(Q_3)=c$. In the future JUNGLE may incorporate Machine Learning (ML) approaches. At that:

(11) $S(Q_3)=a \rightarrow G(Q_i)$.

(12) $S(Q_3)=e \rightarrow G(Q_i)$.

where **a** means "the memorized logical inference is abbreviated"; **e** means *an explanation of the obtained earlier results*. $f(Q_i)<1$ is obtained in the following cases:

(13) $f^p(Q_i)=\max_j(f^p{}_j(Q_i)),j=1,...7 \rightarrow f(Q_i)=0.5\, f^p(Q_i)$.

where $f^p(Q_i)$ are all the memorized evaluations in MOC.

(14) $f^p(Q_i)=\max_t(f^p{}_j(Q_i,t)),j=1,...7 \rightarrow f(Q_i)=0.7\, f^p(Q_i)$.

Only the last remembered value for the corresponding $f(Q_i)$ is taken into account in (14). Some of the above presented procedures are included not only in the FRONTAL, but also in the neighboring methods. The set of all those methods uses the same JUNGLE principle.

The goal function *is easy to change* (see Fig. 1), so the procedures from 1 up to 7 may operate not only with data, but also with goals. E.g. BIND can operate with hypothesis I with hypothesis J in S_5, etc.

3. Experimental Studies and Some of Theoretical Results

The software for the research includes more than 20 programs written in Visual Basic and more than 200 MB Excel data. The assistant and defensive software consists of more than 20 programs in C and C++.

The introduced method generated new results even during the first investigations in 1986. The following strategy was formulated later. The target is to find dependencies in the arrangements of different sets of numbers, e.g. which are multiples of 17. (For example see Fig. 4 and the multiplication cycle 17). One can say that the start is with *zero information*. We introduce descriptions of well known hypotheses, e.g. the twin primes hypothesis, Goldbach's conjecture etc. in the same model. Finally we obtain new mathematical dependencies and formulas. In practice this approach starts with a research of the twin primes hypothesis with a difference of 2: these are couples of prime numbers 5 and 7, 11 and 13 etc. The hypothesis is based on the suggestion that there exist an infinite number of such similar pairs. The hypothesis formalization *must not be mistaken* with the goal function. It is simply a model inside the given sets of progressions. The research of the multiplication operations with prime numbers in different numerical models, e.g. in S_5 leads to the conclusion that the principle properties of different composite numerical unions are also prime number functions (15), (16)! This result at a first glance is very remote from the twin primes hypothesis. This result relates to the proof of Theorem 1 which was not a target in the research. Nevertheless it may assist in the process of solving for many different **goals.** The famous Dirichlet's theorem is a corollary from the Theorem 1.

$$(15)\qquad c_{K,6,1}(x) = \sum_{p=7}^{p_{z7}} c_{K-1,6,1}(\frac{x}{p}) + \sum_{p=5}^{p_{z5}} c_{K-1,6,5}(\frac{x}{p}).$$

$$(16)\qquad c_{K,6,5}(x) = \sum_{p=5}^{p_{z5}} c_{K-1,6,1}(\frac{x}{p}) + \sum_{p=7}^{p_{z7}} c_{K-1,6,5}(\frac{x}{p}).$$

where $p_{za} \in \{a+6k\}^{\infty}_{k=0}$, $C_{k,6,a}(x)$ are all the composites $\leq x$ from $\{a+6k\}^{\infty}_{k=0}$ which contain k prime divisors.

Theorem 1.

We have the interval [0,x]. In it we have two progressions $\{m_1+nk\}^{\infty}_{k=0}$ and $\{m_2+nk\}^{\infty}_{k=0}$ and the relevant numbers are mutually prime: $(m_1,n)=1$, $(m_2,n)=1$. Denote $\Delta\pi_{n,mi}$ (x), i=1,2. The denotation introduces the difference (delta) in the number of the primes $\leq x$ included in both progressions. This difference may not be greater then the number of the primes in the range [0, $\sqrt{x}$], which is signed as follows: $\Delta\pi_{n,mi}(x) \leq \pi(\sqrt{x}\ )$.

The Theorem 1 proof is given in [Jotsov2 1999]. Theorem 1 is the basic tool for the derivation of the twin primes formula:

$$(17)\qquad P_X(p, p+2) \sim 1.320323632\frac{(\pi(x))^2}{x}.$$

where $P_x(p,p+2)$ is the number of twin prime couples $\leq x$, ~ means "asymptotically equal". The solutions below are related to the well known Hardy-Littlewood's hypothesis, the formalization of which is introduced in (19). The formalization check of it revealed a series of inconsistencies, so the hypothesis was transformed in (18). Finally the FRONTAL method has lead to a new hypothesis 1 which is stronger than the Hardy-Littlewood's.

$$(18)\qquad P_X(p, p+d_1,\dots p+d_{z-1}) \geq K_z\frac{(\pi(x))^z}{x^{z-1}}.$$

where P_x is the number of z-tuples $\leq x$. They have different admittable differences between, and K_z are the corresponding coefficients [Riesel 1985].

$$(19)\qquad P_X(p, p+d_1,\dots p+d_{z-1}) \sim K_z\frac{x}{(\ln x)^z}.$$

Hypothesis 1.

Denote **z** the arithmetic progressions $\{a_1+b_1k\}^{\infty}_{k=0}$ ... $\{a_z+b_zk\}^{\infty}_{k=0}$ with a *noncoinciding* step of progressions. Let all the corresponding (a,b)=1. If **z**-tuples of positive integers $(c_i,d_i,\dots z_i)$ are compared; all of them are positive integer numbers; $c_i=a_2-a_1+(b_2-b_1)(i-1+w_1)$ ... $z_i-c_i=a_z-a_1+(b_z-b_1)(i-1+w_z)$, and **w** are positive integers, then there exist infinitely many such **z**-tuples $(c_i, d_i, \dots z_i)$ in which all the numbers are primes $c_i \in P$, $d_i \in P$, ... $z_i \in P$.

Hypothesis 1 is formulated as a result of the application of Theorem 1 to the formula (18). Finally the MOC procedure was applied to the model of the Hardy-Littlewood's hypothesis in S_5. At the end we shall reveal an indicative fact. The paper containing the draft with the Theorem 1 proof is one page long. The initial version of the theorem comprised more than 30 pages with several bottlenecks. The author improved the proof using manually the FRONTAL method and the CROSSWORD method. The obtained by now results confirm the effect in cases with *infinite* sets of integers and they reveal possibilities for solving problems with *higher complexity*.

4. Some of the Advantages

The greater part of the seven procedures and their interaction inside the FRONTAL method are completely original. This method operates in the *environment of other methods* which are also proposed by the same author. The usage of this method in Number Theory leads to new mathematical results which are widely discussed and acknowledged as original. Part of them is accepted for a publication in Australia. Another fraction is under consideration in AMS. The results from section 3 after Theorem 1 are only partially issued in the math periodicals. They are presented as an illustration of the method for the way in which a **front** of mutually related solutions can be formed. It is possible to set a way for applications of contemporary IT in computational mathematics, residing on the presented method.

5. Conclusions

A new IT method is proposed for the interactive construction of formulas and proofs in Number Theory. It follows from the consideration that *even a non-specialist* can make easy explainable solutions if she/he uses the present work with the described method. The method is multi-target oriented and its main part is domain independent.

Bibliography

[Altshuller 1979] G. S. Altshuller, *Creation as an Exact Science*, Moscow, Sov. Radio, 1979.

[Baldi 1995] P. Baldi, “Gradient learning algorithm overview: A general dynamical systems perspective,” *IEEE Trans. on Neural Networks*, Vol. 6, pp. 182-195, Jan. 1995.

[EAEA 1997] *Evolutionary Algorithms in Engineering Applications*, D. Dasqupta and Z. Michalewicz (Eds.), Springer, Berlin etc., 1997.

[EA 1997] *Evolutionary Algorithms*, L. D. Davis et al. (Eds.), Springer - Verlag, New York etc., 1999.

[Hadamard 1975] J. Hadamard, Essai sue la Psychologie de l'Invention dans le Domaine Mathematique, Paris, Gautier-Villas, 1975.

[Jotsov1 1999] V. Jotsov, “On one approach in proof search and refinement,” *J. Artif. Intell.* (National Academy of Sciences - Ukraine), No. 2, pp. 97-103, 1999.

[Jotsov2 1999] V. Jotsov, “On the usage of discovering systems' approaches in Number Theory,” *IIT Working Paper* WP/83B, Bulg. Acad. of Sciences, 14pp., December 1999.

[Lenat 1976] B. Lenat, “AM: an artificial intelligence approach to discovery in mathematics as heuristic search,” *Memo* AIM-286, Stanford University, CA, 1976.

[Lenat 1983] D. B. Lenat, “Why AM and EURISKO appear to work,” *Artif. Intell.*, Vol. 21, pp. 61-98, 1983.

[Narin'yani 2000] A. S. Narin'yani et. al., “Constraint programming and subdefinite models,” preliminary version.

[Nordin 1999] P.Nordin and A. Eriksson, M. Nordahl, “Genetic reasoning: evolutionary induction of mathematical proofs,” in R. Poli, P. Nordin, W. B. Langton, T. C. Fogarty (Eds.), Genetic Programming, *LNCS* 1598, Springer, Berlin etc., pp. 221-231, 1999.

[Polya 1963] G. Polya, *How to Solve It*, Princeton University Press, N.J., 1963.

[Riesel 1985] H. Riesel, Prime Numbers and Computer Methods for Factorization, Birkhauser, Boston, 1985.

[Shvetsov 1997] I. E.Shvetsov, V. V. Telerman, and D. M. Ushakov, “NeMo+: object-oriented constraint programming environment based on subdefinite models,” Proc. Third Intern. Conf. on Principles and Practice of Constraint Programming (CP'97), *LNCS* 1330, Springer, Berlin etc., pp. 373-385, 1997.

Author information

Vladimir S. Jotsov – College of Library Science Education and Information Technologies (KBIT); Intstitute of Information Technologies – Bulgarian Academy of Sciences (IIT-BAS); 1113 Sofia, P.O. Box 161, Bulgaria; tel. 359 2 9708592, e-mail: jotsov@ieee.org

ON STATISTICAL HYPOTHESIS TESTING VIA SIMULATION METHOD

B. Dimitrov, D. Green, Jr., V.Rykov, P. Stanchev

Abstract: A procedure for calculating critical level and power of likelihood ratio test, based on a Monte-Carlo simulation method is proposed. General principles of software building for its realization are given. Some examples of its application are shown.

1 Introduction

In this paper we show how the present day fast computer could solve non-standard old statistical problems. In most cases statisticians work with approximations of test statistics distributions, and then use statistical tables. When approximations do not work the problem is usually tabled. We propose a simulation approach which we do believe could be helpful in many cases

The problem of statistical hypothesis testing is very important for many applications. In the notable but rare case, it is possible to find some simple test statistic having a standard distribution. However, in the general case the statistics based on the *Likelihood Ratio Test* (LRT) does not usually have one of the known standard distributions. The problem could be overcome with the help of an appropriate simulation method. This method was first used in [3] for a specific case of almost lack of memory (ALM) distributions. In this paper we propose a general approach for using the method, describe its general principles and algorithms, show how to build up an appropriate software, and illustrate with examples its application.

2 LRT and the Simulation Approach

It is well known according to Neyman-Pearson theory [7], that the *most powerful test* for testing a null hypothesis H_0 : *f(x)* = f_0 *(x)* versus an alternative H_1 : *f(x)* = f_1 *(x)* is the LRT. For this test, the *critical region W* for a sample x_1 ,..., x_n of size *n* has the form

$$W=\left\{ (\mathrm{x}_1, \ldots, \mathrm{x}_\mathrm{n}) : \mathrm{w}(\mathrm{x}_1, \ldots, \mathrm{x}_\mathrm{n}) = \frac{f_1(\mathrm{x}_1, \ldots, \mathrm{x}_\mathrm{n})}{f_o(\mathrm{x}_1, \ldots, \mathrm{x}_\mathrm{n})} > \mathrm{t} \right\},$$

where $f_0(\mathrm{x}_1,\ldots,\mathrm{x}_\mathrm{n})$ and $f_1(\mathrm{x}_1,\ldots,\mathrm{x}_\mathrm{n})$ are joint probability densities of the distributions of observations (the *likelihood functions*) under hypotheses H_0 and H_1 with probability density functions (p.d.f.) $f_0(.)$ and $f_1(.)$ respectively. The notation

$$\mathrm{w}(\mathrm{x}_1,\ldots,\mathrm{x}_\mathrm{n}) = \frac{f_1(\mathrm{x}_1,\ldots,\mathrm{x}_\mathrm{n})}{f_o(\mathrm{x}_1,\ldots,\mathrm{x}_\mathrm{n})}$$

is used for *test's statistic*. For independent observations this statistic can be represented in the form

$$\mathrm{w}(\mathrm{x}_1, \ldots, \mathrm{x}_\mathrm{n}) = \frac{f_1(\mathrm{x}_1, \ldots, \mathrm{x}_\mathrm{n})}{f_0(\mathrm{x}_1, \ldots, \mathrm{x}_\mathrm{n})} = \prod_{1 \le i \le n} \frac{f_1(x_i)}{f_0(x_i)}$$

Considering the observations x_1 ,..., x_n as independent realizations of random variable (i.i.d. r.v.) *X* with p.d.f. $f_0(.)$ the *significance level* of the test is

$$\alpha = P_{H_0}\{W\} = P_{H_0}\{w(X_1,\ldots,X_n) > t_\alpha\}. \qquad (1)$$

Here an appropriate *critical value* t_α for any given significance level α is the smallest solution of equation (1).

On the other hand, considering the same observations x_1 ,..., x_n as independent realizations of random variable *Y* with p.d.f. $f_1(.)$, *the power of the test* is

$$\pi_\alpha = P_{H_1}\{W\} = P_{H_1}\{w(Y_1,\ldots,Y_n) > t_\alpha\}. \qquad (2)$$

Thus, to find the critical value for a given significance level α and the power of the test $\pi\ (\alpha)$, a statistician needs to know the distributions of the test statistic w under hypotheses H_0 and H_1 .

For parametric hypothesis testing the problem becomes more complicated because in such cases one has to be able to find a free of parameter distribution of this statistic.

To avoid calculations of these functions we propose to use the simulation method. This means that instead of searching for exact statistical distributions, we will calculate appropriate empirical distributions as their estimations. This method gives the desired results due the fact (based on the *Strong Law of Large Numbers*) that the empirical distribution function of the test statistic converges with probability one to the theoretical distribution.

In the following, due to numerical reasons, instead of statistic *w* we will use its natural logarithm, and for simplicity we will denote this statistic with the same letter, w,

$$w = \ln \prod_{1 \le i \le n} \frac{f_1(x_i)}{f_0(x_i)} = \sum_{1 \le i \le n} (\ln f_1(x_i) - \ln f_0(x_i)) . \tag{3}$$

Due to additional statistical reasons, instead of the cumulative distribution functions (CDF) of the statistic w under hypotheses H_0 and H_1 , we will use their tails,

$$\overline{F_o}\ (t) = P_{H_0}\{w(X_1,\ldots,X_n) > t\ \}, \tag{4}$$

and

$$\overline{F_1}\ (t) = P_{H_1}\{w(Y_1,\ldots,Y_n) > t\ \}. \tag{5}$$

For large size samples, *n*>>1, it is possible to use a simplier approach based on the *Central Limit Theorem*. It is well known that this theorem provides a normal approximation of the distribution for sums of i.i.d. r.v.'s under conditions of existence of finite second moments. This would allow one to calculate and use only two moments of the test statistic w and then to calculate the appropriate significance level and power of the test making use of the respective normal approximation.

To show how it works, let us denote by *U* and *V* the r.v.'s

$$U = \ln f_1(X) - \ln f_0(X), \qquad V = \ln f_1(Y) - \ln f_0(Y),$$

where *X* and *Y* are taken from distributions with densities $f_0(.)$ and $f_1(.)$ respectively, corresponding to hypotheses H_0 and H_1 . Denote by μ_U , μ_V and σ_U^2 , σ_V^2 their expectations and variances respectively, when they exist. Then, for large samples, *n*>>1, under null hypothesis, the test's statistic w has approximately normal distribution with parameters $n\,\mu_U$, and $n\,\sigma_U^2$. This means that the significant level t_α for given value of α can be found from the equation

$$\alpha = P_{H_0}\{w(X_1,\ldots,X_n) > t\ \} = P_{H_0}\{\frac{w - n\mu_U}{\sigma_U\sqrt{n}} > \frac{t_\alpha - n\mu_U}{\sigma_U\sqrt{n}}\} \approx 1 - \Phi\left(\frac{t_\alpha - n\mu_U}{\sigma_U\sqrt{n}}\right),$$

or equivalently

$$\frac{t_\alpha - n\mu_U}{\sigma_U\sqrt{n}} \approx z_{1-\alpha} .$$

Here $z_{1-\alpha}$ is the (1-α)-quantile of the standard normal distribution. Thus, the critical value t_α for the test statistic w at a given significance level α is

$$t_\alpha \approx n\,\mu_U + z_{1-\alpha}\ \sigma_U\sqrt{n} . \tag{6}$$

The power of the test equals

$$\pi_\alpha = P_{H_1}\{\, w(Y_1,\ldots,Y_n) > t_\alpha \} = P_{H_1}\{\frac{w - n\mu_V}{\sigma_V\sqrt{n}} > \frac{t_\alpha - n\mu_V}{\sigma_V\sqrt{n}}\}$$

$$= 1 - \Phi\left(\frac{t_\alpha - n\mu_V}{\sigma_V\sqrt{n}}\right) = 1 - \Phi\left(\frac{\mu_U - \mu_V}{\sigma_V}\sqrt{n} + z_{1-\alpha}\frac{\sigma_U}{\sigma_V}\right). \tag{7}$$

From this equality it is possible to see that the power of the test mainly depends on the difference in expectations of the r.v.'s *U* and *V*.

In some cases the parameters μ_U, μ_V and σ_U^2, σ_V^2 can be calculated in a closed (explicit) form. In general it is possible to estimate them also with the help of Monte-Carlo techniques and then use the respective estimated values instead of the exact ones. Appropriate algorithms for calculating the empirical cumulative distribution functions (CDF) of the test's statistic under hypotheses H_0 and H_1 for both cases are described below.

3 Algorithms

In this section two algorithms for calculation of the tails of CDF of LRT's statistic w under both null and alternative hypothesis (the null H_0 : *f(x)* = f_0 *(x)* and the alternative H_1 : *f(x)* = f_1 *(x)*), based on a Monte-Carlo method are proposed. One algorithm can be applied for any sample size *n*. The second algorithm should be used for large samples, *n*>>1, mainly when the parameters μ_U, μ_V and σ_U^2, σ_V^2 are finite.

Algorithm 1. LRT for any sample size

Begin. Select the p.d.f.'s $f_0(.)$ and $f_1(.)$, and the sample size *n*.

Step 1. Generate a sequence of *N* random samples $(x_1^{(j)},\dots,x_n^{(j)})$, *j=1,…,N,* from a distribution with p.d.f. $f_0(.)$, and calculate *N* values of the test statistics

$$w_j = w\,(x_1^{(j)},\dots,x_n^{(j)}) = \sum_{1\le i\le n}(\ln f_1(x_i^{(j)}) - \ln f_0(x_i^{(j)})),\ j=1,\dots,N. \quad (8)$$

Step 2. Calculate the complementary empirical distribution function

$\overline{F}_{0,N}(t) = \frac{1}{N}\{$ number of $w_j's > t\}$, $t > 0$.

Step 3. Calculate the critical value t_α for the test statistic w at a given significance level α as the smallest solution of the equation $\overline{F}_{0,N}(t) = \alpha$.

Step 4. Generate a sequence of *N* random samples $(y_1^{(j)},\dots,y_n^{(j)})$,*j=1,…,N,* from a distribution with p.d.f. $f_1(.)$, and calculate the values of the test statistics w_j, analogous to (8), with $y_i^{(j)}$'s instead of $x_i^{(j)}$'s.

Step 5. Calculate the complementary empirical distribution function for the new sample

$\overline{F}_{1,N}(t) = \frac{1}{N}\{$number of $w_j's > t\}$, $t > 0$.

Step 6. Calculate the power of the test statistic w at the given significance level α from the equation

$\overline{F}_{1,N}(t_\alpha) = \pi_\alpha$.

Step 7. **Enter the application's data:** For a given user's sample $(x_1,\dots,x_n)$, calculate the test statistic

$$w = w\,(x_1,\dots,x_n) = \sum_{1\le i\le n}(\ln f_1(x_i) - \ln f_0(x_i)).$$

Calculate the *p*-value for testing the null hypothesis H_0 : *f(x)* = f_0 *(x)* versus the alternative H_1 : *f(x)* = f_1 *(x)* by making us of the Likelihood Ratio Test from the equation

$\overline{F}_{0,N}(w) = p -$value.

Make a decision by comparing the calculated *p*-value and α. Alternatively, reject the hypothesis H_0 if the inequality $w > t_\alpha$ holds.

Calculate the *probability of committing an error of type II* (when testing the null hypothesis H_0 : *f(x)* = f_0 *(x)* versus the alternative H_1 : *f(x)* = f_1 *(x)* by making use of the Likelihood Ratio Test by the simulation method) from the equation

$1-\overline{F}_{1,N}(w)=\beta -$ the probability of type II error.

Step 8. Print results:

The chosen null hypothesis H_0 : *f(x)* = f_0 *(x)* and alternative hypothesis H_1 : *f(x)* = f_1 *(x),* the selected significance level α, and the sample size *n* .

- The *p*-value of the test;
- The power of the test, $\pi_\alpha = 1-\beta$;
- The calculated value of the test statistic w, and the calculated by simulation critical value t_α ;
- The graphs of the tails of the empirical CDFs $\overline{F}_{0,N}(t)$ and $\overline{F}_{1,N}(t)$.

End.

For large size samples when the second moments of the r.v.'s *U,* and *V* exist, it is possible to modify and simplify the simulation algorithm as shown below.

Algorithm 2. LRT for large samples.

Begin. Select the p.d.f.'s $f_0(.)$ and $f_1(.)$, and the sample size *n*.

Step 1. Generate a sequence of *N* random variables $(x_1,\dots,x_N)$, from a distribution with p.d.f. $f_0(.)$, and calculate *N* values of the statistics

$$u_j = u\,(x_j) = \ln f_1(x_j) - \ln f_0(x_j),\quad j=1,\dots,N. \qquad (9)$$

and its sample mean $\overline{u}$, and sample variance s_u^2 according to

$$\overline{u} = \frac{1}{N}\sum_{1\le j\le N} u_j,\quad s_u^2 = \frac{1}{N}\sum_{1\le j\le N}(u_j-\overline{u})^2 = \frac{1}{N}\sum_{1\le j\le N}u_j^2-(\overline{u})^2. \qquad (10)$$

Step 2. Calculate the critical value t_α for the test statistic w at a given significance level α from the equation

$$t_\alpha \approx n\overline{u} + z_{1-\alpha}\cdot s_u\cdot\sqrt{n}, \qquad (11)$$

where $z_{1-\alpha}$ is the (1-α)-quantile of the standard normal distribution.

Step 3. Generate a sequence of *N* random variables $(y_1,\dots,y_N)$, from a distribution with p.d.f. $f_1(.)$, and calculate *N* values of the statistics

$$v_j = v\,(y_j) = \ln f_1(y_j) - \ln f_0(y_j),\quad j=1,\dots,N. \qquad (12)$$

and its sample mean $\overline{v}$, and sample variance s_v^2 according to (10) for the data (12).

Step 4. Calculate the power of the test at the given significance level α from the equation

$$\pi_\alpha = 1-\Phi\left(\frac{\overline{u}-\overline{v}}{s_V}\sqrt{n}+z_{1-\alpha}\frac{s_U}{s_V}\right). \qquad (13)$$

where $\Phi(x)$ is the c.d.f. of the standard normal distribution.

Step 5. **Enter the application's data:** For a given user's sample $(x_1,\dots,x_n)$, calculate the test statistic

$$w = w\,(x_1,\dots,x_n) = \sum_{1\le i\le n}(\ln f_1(x_i) - \ln f_0(x_i)).$$

Calculate the *p*-value for testing the null hypothesis H_0 : *f(x)* = f_0 *(x)* versus the alternative H_1 : *f(x)* = f_1 *(x)* by making us of the Likelihood Ratio Test from the equation

p-value= $P_{H_0}\{w(X_1,\dots,X_n) > w\} \approx 1-\Phi\left(\frac{w-n\cdot\overline{u}}{s_u\sqrt{n}}\right),$

where w is the calculated statistic from the sample. Make a decision by comparing the calculated *p*-value and α. Alternatively, reject the hypothesis H_0 if the inequality $w > t_\alpha$ holds, where t_α is calculated by (11).

Calculate the *probability of committing an error of type II* (when testing the null hypothesis H_0 : *f(x)* = f_0 *(x)* versus the alternative H_1 : *f(x)* = f_1 *(x)* by making use of the LRT by the simulation method) from the equation β = $1-\pi_\alpha$ with the π_α calculated in Step 4.

Step 6. Print results:

- The chosen null hypothesis H_0 : *f(x)* = f_0 *(x)* and alternative hypothesis H_1 : *f(x)* = f_1 *(x),* the selected significance level α, and the sample size *n* ;
- The *p*-value of the test;
- The power of the test, $\pi_\alpha = 1-\beta$;
- The calculated test statistic w , and the calculated by simulation critical value t_α ;
- The graphs of the tails of the CDFs

$$\overline{F}_{0,N}(t) = 1 - \Phi\left(\frac{t-n\cdot\bar{u}}{s_u\sqrt{n}}\right), \text{ and } \overline{F}_{1,N}(t) = 1 - \Phi\left(\frac{t-n\cdot\bar{v}}{s_v\sqrt{n}}\right).$$

End.

4 The Software

For practical application of the above algorithms an appropriate software should be utilized. The software should have a friendly interface, which allows work in two different regimes: *individual* (customized), and *automatic.*

In the individual regime only particular observations are tested for any pair of given null and alternative hypotheses. Automatic regime allows one to calculate and show the significance level and power functions as functions of the test's statistic, and also as functions of some parameters of the model. In this way it would allow one to investigate some parametric models.

The interface includes the main menu, which allows the users to choose:

- the regime for investigation;
- the p.d.f. for hull and alternative hypotheses from a given list of distributions, which include almost all standard discrete and continuous distributions, or
- propose an option to the user for selecting probability distribution’s formula or tables of his/her own choice.

The submenu allows:

- one to choose the parameter values for hypothesis testing for individual regime; or
- one to choose the intervals and steps of increment for parameters varying for the problem investigated in an automatic regime.

The software allows also different type of presentation of the results: numerical, graphical, comparison with respect to various variables, or with respect to family of functions. These and other appropriate possibilities make the content of the design menu.

The design will be based on the new technologies presented in [8].

5 An Example

Below we consider one example on which the work of algorithms in the previous section will be illustrated.

Example. An ALM distribution versus other ALM distribution with uniform distribution.

It is known that when in the ALM distribution

$$f(x) = (1-a)\cdot a^{\left[\frac{x}{c}\right]} f_Y\left(x - \left[\frac{x}{c}\right]c\right), \qquad (14)$$

where a is a parameter of distribution, c is the length of a period, and $f_Y(x)$ is an arbitrary distribution on the interval [0,c). More details about ALM distributions can be found in [5].

Here for the ALM distribution in the null hypothesis H_0 we choose $f_0(x),$ presented by (14) with parameters chosen in the following way

$$c=1, \quad a_0=.5, \quad f_{Y,0}(x)=1 \text{ for } 0 \le x \le 1. \tag{15}$$

This means that the r.v. *X* with distribution (14) is based on the uniform distribution of Y_0 on [0,1] (cycle of length 1), and probability for jump over a cycle without success is a_0 =.5. Any other choice of the parameter $a_0 \neq .5$ will produce an ALM distribution $f_1(x),$ different from the chosen $f_0(x)$. And this p.d.f. $f_1(x)$ will appear in our considerations as an alternative hypothesis H_1 .

Thus, we study the likelihood ratio test according to Algorithms 1 and 2 above with the choice for the p.d.f. $f_0(x)$, with a_0 =.5, and choosing various other values for parameter $a_1 \neq .5$. In studying the power function dependence on significance level α we select a_1 =.05,.1,.15,…,.9,.95; *N* = 10000, *n* = 10, c = 1.

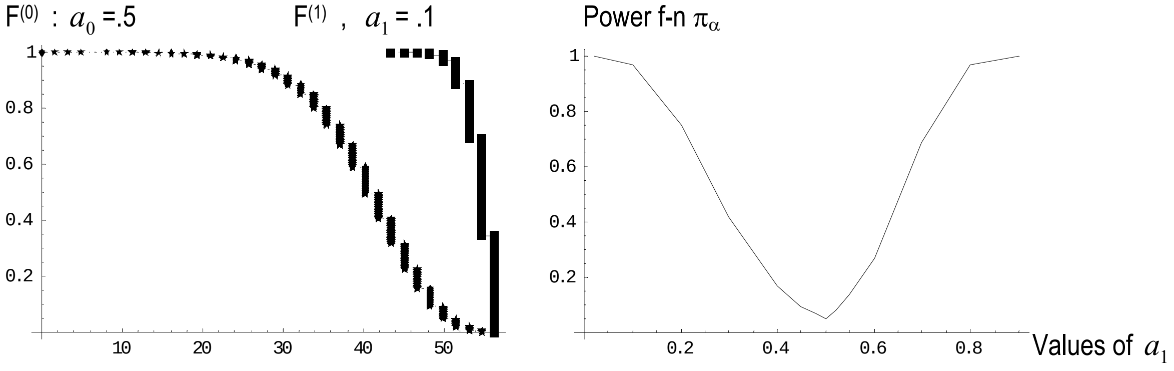


Fig. 1. Cumulative distribution functions for the test statistic and the power function of the test

The results for the power function in this case of significance level α=.05 are shown on Fig. 1.

6 Conclusions

The problem of hypotheses testing arises in many statistical applications. In analytical form its solution can be done for a very limited number of cases. The method proposed in this paper gives the solution for practically all cases. Nevertheless, for its practical realization special computer tools with friendly interface are needed. This work is now in the progress, and we show here some examples of the approach used for some special case of distributions, - so called almost lack of memory distributions.

References

[1] S.Chukova, and B.Dimitrov (1992) On distributions having the almost lack of memory property. *Journal of Applied Probability*, 29 (3), 691-698.

[2] S.Chukova, B.Dimitrov, and Z. Khalil (1993) A characterization of probability distributions similar to the exponential. *The Canadian Journal of Statistics*, 21 (3), 269-276.

[3] B.Dimitrov, Z.Khalil, M.Ghitany, and V.Rykov (2001) Likelihood Ratio Test for Almost Lack of Memory Distributions. *Technical Report No. 1/01*, November 2001, Concordia University, 2001.

[4] B.Dimitrov, and Z.Khalil (1992) A class of new probability distributions for modeling environmental evolution with periodic behavior. *Environmetrics*, 3 (4), 447-464.

[5] B.Dimitrov, S.Chukova, and D.Green Jr.(1997) Distributions in Periodic Random Environment and Their Applications. *SIAM J.on Appl. Math.*, 57 (2), 501-517.

[6] B.Dimitrov, Z.Khalil, and L.Le Normand (1993) A statistical study of periodic environmental processes, *Proceedings of the International Congress on Modelling and Simulation*, Dec. 6-10, 1993, The Univ. of Western Australia, Perth, v. 41, pp. 169 - 174.

[7] R.A.Johnson (1994) *Miller & Freund's Probability and Statistics for Engineers,5th edition,* Prentice Hall, New Jersey 07632.

[8] J.Carroll (2002) *Human - Computer Interaction in the New Millennium*, ACM Press, NY.

[9] A,M.Law and W.D.Kelton (2000) *Simulation Modeling and Analysis,* 3d edition, McGraw-Hill, N.Y.

Author Information

B. Dimitrov - bdimitro@kettering.edu
D. Green, Jr. - dgreen@kettering.edu
V.Rykov - vrykov@kettering.edu
P. Stanchev - pstanche@kettering.edu
Department of Science & Mathematics, Kettering University, Flint, Michigan, USA
1700 West Third Avenue, Flint, Michigan 48504-4898, USA

A GRADIENT-TYPE OPTIMIZATION TECHNIQUE FOR THE OPTIMAL CONTROL FOR SCHRODINGER EQUATIONS

M. H. FARAG

Abstract: In this paper, we are considered with the optimal control of a schrodinger equation. Based on the formulation for the variation of the cost functional, a gradient-type optimization technique utilizing the finite difference method is then developed to solve the constrained optimization problem. Finally, a numerical example is given and the results show that the method of solution is robust.

Keywords: Optimal control, schrodinger equation, Existence and uniqueness theory, Gradient method.

AMS subject classification: 49J20, 49M29, 49M30, 49K20

1. Introduction

Optimal control of systems governed by partial differential equations is an application-driven are of mathematics involving the formulation and solution of minimization problems [1,3]. In this paper, we are considered with the optimal control of a schrodinger equation. Based on the formulation for the variation of the cost functional, a gradient-type optimization technique utilizing the finite difference method is then developed to solve the constrained optimization problem. Finally, a numerical example is given and the results show that the method of solution is robust.

2. Problem Formulation

We consider the functional on the form

$$(1)\quad J(u) = \alpha_0 \int_0^T \left|y(0,t) - f_0(t)\right|^2 dt + \alpha_1 \int_0^T \left|y(l,t) - f_1(t)\right|^2 dt$$

which is to minimized under the conditions

$$(2)\quad i\frac{\partial y}{\partial t} + B_0 \frac{\partial^2 y}{\partial x^2} - u\,y = f(x,t)\,,\ (x,t)\in \Omega=(0,l)\ x\ (0,T)$$

$$(3)\quad y(x,0) = 0\,,\ x \in (0,l)$$

$$(4)\quad \frac{\partial\, y(0,t)}{\partial\, x} = \frac{\partial\, y(l,t)}{\partial\, x} = 0\,,\ t \in (0,T)$$

over the class

$$U=\left\{u : u(x,t) \in W_2^{0,1}(\Omega),\ \alpha_0 \le u(x,t) \le \alpha_1\,, \left|u_t\right| \le \alpha_2\ , \forall\ (x,t) \in \Omega\right\}$$

where $\alpha_k \ge 0\,, k=\overline{0,2}\,,\ \alpha_1 + \alpha_2 \ne 0\,, l, T, B_0 > 0$ are given numbers and $f_0(t),\ f_1(t) \in W_2^1(0,T), \varphi(x) \in W_2^1\ (0,1)$, are given functions.

Definition 1.

The problem of finding the function $y(x,t) \in V_2^{0,1}(\Omega)$ from condition (2)-(4) at given $u \in U$ is called the reduced problem.

Definition 2.

A function $y(x,t) \in V_2^{0,1}(\Omega)$ is said to be a solution of the problem (2)-(4), if for all $\eta = \eta(x,t) \in W_2^{1,1}(\Omega)$ the equation

$$(5)\ \int_\Omega [-iy\frac{\partial\eta}{\partial t} - B_0 \frac{\partial y}{\partial x}\frac{\partial \eta}{\partial x} - uy\overline{\eta}]\,dx\,dt$$

$$= \int_\Omega f(x,t)\overline{\eta}\,dx\,dt + i\int_0^1 \varphi\ \overline{\eta}(x,0)\,dx$$

is valid and $\eta(x,T) = 0,$ but $\overline{\eta}$ is the adjoint of η .

Proposition 1

Let $f(x,t) \in W_2^{0,1}(\Omega)$ and $\varphi(x,t) \in W_2^1\ (0,1)$. Then the problem (2)-(4) has a unique solution and satisfies the following estimate

$(6)\ \|y\|^2_{V_2^{1,0}(\Omega)} \le C_1 [\|\varphi\|^2_{W_2^1(0,1)} + \|f\|^2_{W_2^{0,1}(0,1)}]$ is valid and $C_1 > 0$ is dos not depend on φ and f .

Proposition 2

Let $\varphi(x,t) \in W_2^2\ (0,1)$. Then the solution of the reduced problem (2)-(4) $y(x,t) \in V_2^{0,1}(\Omega)$ belongs to the space $W_2^{2,1}(\Omega)$ and satisfies the following estimate

$(7)\ \|y\|^2_{W_2^2(\Omega)} + \|y_t\|^2_{L_2(0,1)} \le C_2 [\|\varphi\|^2_{W_2^2\ (0,1)} + \|f\|^2_{W_2^{0,1}(\Omega)}]$ is valid and $\forall\, t \in [0,T], C_2 > 0$ is dos not depend on φ and f .

Proposition 3

Let all the conditions Proposition 2 be valid. Then the optimal control problem (1)-(4) has at least one solution.

3. Variation of the Cost Functional

3.1 The Adjoint Problem

Results [4] imply that the function $\Phi = \Phi(x,t,u)$ is a solution in $L_2\ (\Omega)$ of the adjoint problem

$$(8)\qquad i\frac{\partial\Phi}{\partial t} + B_0\frac{\partial^2\Phi}{\partial x^2} - u\Phi = 0\ , (x,t) \in \Omega = (0,1)\ x\ (0,T)$$

$$\Phi(x,T) = 0, \quad x \in (0,1)$$

$$(9) \qquad \frac{\partial \Phi(0,t)}{\partial x} = -\frac{2\alpha_0}{B_0}[\, y(0,t) - f_0(t)\,], \quad t \in (0,T)$$

$$\frac{\partial \Phi(1,t)}{\partial x} = \frac{2\alpha_1}{B_0}[\, y(1,t) - f_1(t)\,], \quad t \in (0,T)$$

where $y(x,t)$ is the solution of (1)-(4) corresponding to $u \in U$.

Definition 3.

For each $u \in U$, a function $\Phi(x,t;u)$ is a solution of the adjoint problem (8)-(9) belonging to the control u iff

(I) $\Phi(x,t;u) \in L_2(\Omega)$,

(II) The integral identity

$$(10) \quad \int_\Omega \Phi \left[i \frac{\partial \overline{\eta_1}}{\partial t} + B_0 \frac{\partial^2 \overline{\eta_1}}{\partial x^2} - u\, \overline{\eta_1} \right] dx\, dt$$

$$= -2\alpha_1 \int_0^T [\, y(1,t) - f_1(t)\,]\, \overline{\eta_1}(1,t)\, dt$$

$$+ 2\alpha_0 \int_0^T [\, y(0,t) - f_0(t)\,]\, \overline{\eta_1}(0,t)\, dt$$

is valid $\forall\ \eta_1 \in W_2^{2,1}(\Omega)$, $\eta_1(x,0) = (\eta_1)_x|_{x=0} = (\eta_1)_x|_{x=1} = 0.$

On the basis of the above assumptions and the results [5], we have the following proposition:

Proposition 4.

The adjoint problem (8)-(9)has a unique solution from $L_2(\Omega)$ and he following estimate

$$(11) \quad \|\Phi\|^2_{L_2(\Omega)} \le C_3 [\, \Gamma_1 + \Gamma_2\,], \qquad \text{where}$$

$$\Gamma_1 = \|y(0,t) - f_0(t)\|^2_{W_2^{\frac{1}{2}}(0,T)}, \quad \Gamma_1 = \|y(1,t) - f_1(t)\|^2_{W_2^{\frac{1}{2}}(0,T)}$$

is valid and c_3 is a certain constant.

3.2 The Gradient Formulae of Cost Functional

The sufficient differentiability conditions of the functional (5) and its gradient formulae will be given as follows:

Theorem 1.

Let the above assumptions be satisfied. Then $J(u)$ is Gato differentiable, and its gradient satisfies

$$(12) \qquad \delta J(u) = -\int_\Omega \mathrm{Re}(y\,\Phi)\,\omega\; dx\, dt\,, \qquad \forall\ \omega \in W_\infty^{0,1}(\Omega).$$

Proof :

Suppose that $u \in U$ and $\delta u \in W_\infty^{0,1}(\Omega)$ such that $u + \delta u \in U$ and denoting $\delta y(x,t) = y(x,t;u+\delta u) - y(x,t;u).$ Then $\delta y(x,t;\delta u)$ is the solution of the boundary value problem:

$$(13) \quad i\frac{\partial \delta y}{\partial t} + B_0 \frac{\partial^2 \delta y}{\partial x^2} - (u+\delta u)\ \delta y = y(x,t)\ \delta u,\ (x,t) \in \Omega,$$

$$(14) \quad \delta y(x,0)=0,\ x \in (0,l),\ \frac{\partial\, \delta y(0,t)}{\partial x} = \frac{\partial \delta y(l,t)}{\partial x} = 0,\ t \in (0,T)$$

and the solution of the above boundary value problem satisfies the following estimation

$$(15) \quad \| \delta y \|^2_{W_2^{2,0}(\Omega)} \le C_4 \| \delta u\ y \|^2_{W_2^{0,1}(\Omega)}$$

where C_4 is a constant and independent of δu .

From (15) and using the theorem of imbedding [6], we have

$$(16) \quad \| \delta y(0,t) \|_{L_2(0,T)} + \| \delta y(l,t) \|_{L_2(0,T)} \le C_5 \| \delta u\ y(x,t) \|_{W_2^{0,1}(\Omega)}$$

where C_5 is a constant and independent of δu.

The increment of the functional $J(u)$ can be expressed as:

$$\begin{aligned}(17) \quad \delta J &= J(u+\theta u) - J(u) \\ &= 2\alpha_1 \operatorname{Re} \int_0^T [\, y(l,t) - f_1(t)\,]\ \overline{\delta y}(l,t)\ dt \\ &\quad + 2\alpha_0 \operatorname{Re} \int_0^T [\, y(0,t) - f_0(t)\,]\ \overline{\delta y}(0,t)\ dt \\ &\quad + \alpha_1 \| \delta y(l,t) \|^2_{L_2(0,T)} + 2\alpha_0 \| \delta y(0,t) \|^2_{L_2(0,T)}\end{aligned}$$

If we take complex adjoint for (10),(13), we have

$$\begin{aligned}(18) \quad &\int_\Omega \overline{\Phi}\, [\, i\frac{\partial \eta_1}{\partial t} + B_0 \frac{\partial^2 \eta_1}{\partial x^2} - u\ \eta_1 \,]\ dx\ dt \\ &= -2\alpha_1 \int_0^T [\, \overline{y}(l,t) - \overline{f}_1(t)\,]\ \eta_1(l,t)\ dt \\ &+ 2\alpha_0 \int_0^T [\, \overline{y}(0,t) - \overline{f_0}(t)\,]\ \eta_1(0,t)\ dt\end{aligned}$$

$$\begin{aligned}(19) \quad &\int_\Omega [\, i\frac{\partial \overline{\delta y}}{\partial t} + B_0 \frac{\partial^2 \overline{\delta y}}{\partial x^2} - (u+\delta u)\ \overline{\delta y}\,]\ \eta\, dx\ dt \\ &= \int_\Omega \overline{y}(x,t)\ \delta u\ \eta\ dx\ dt,\end{aligned}$$

Subtracting (13) from (19), (10) from (18) and in the obtained relation we put Φ, δy instead of η , η_1 , then we have

$$(20)\quad 2\,\alpha_1\ \mathrm{Re}\int_0^T [\,y(1,t)-f_1(t)\,]\,\overline{\delta y}(1,t)\,dt$$

$$+\,2\,\alpha_0\ \mathrm{Re}\int_0^T [\,y(0,t)-f_0(t)\,]\,\overline{\delta y}(0,t)\,dt$$

$$=-\frac{1}{2}\int_\Omega [\,\delta u\,\Phi\,\overline{y}+\delta u\,y\,\overline{\Phi}\,]\,dx\,dt$$

$$-\frac{1}{2}\int_\Omega [\,\delta u\,\delta y\,\overline{\Phi}+\delta u\,\Phi\,\overline{\delta y}\,]\,dx\,dt$$

$$=-\,\mathrm{Re}\int_\Omega y\,\overline{\Phi}\,\delta u\,dx\,dt-\mathrm{Re}\int_\Omega \delta y\,\overline{\Phi}\,\delta u\,dx\,dt\,.$$

By substituting the last relation in (17), we have

$$(21)\quad \delta J=-\,\mathrm{Re}\int_\Omega y\,\overline{\Phi}\delta y\,\delta u\,dx\,dt\ -\mathrm{Re}\int_\Omega \delta y\,\overline{\Phi}\,\delta u\,dx\,dt.$$

$$+\ \alpha_0\,\|\delta y(0,t)\|^2_{L_2(0,T)}+\alpha_1\,\|\delta y(1,t)\|^2_{L_2(0,T)}$$

Suppose that

$$(22)\quad R_1=\alpha_0\,\|\delta y(0,t)\|^2_{L_2(0,T)}+\alpha_1\,\|\delta y(1,t)\|^2_{L_2(0,T)}$$

$$(23)\quad R_2=-\,\mathrm{Re}\int_\Omega \delta y\,\overline{\Phi}\,\delta u\,dx\,dt\,.$$

It is clear that,

$$(24)\quad |R_1|\le\alpha_0\,\|\delta y(0,t)\|^2_{L_2(0,T)}+\alpha_1\,\|\delta y(1,t)\|^2_{L_2(0,T)}.$$

From the formulae of R_2 ,it is estimated as

$$(25)\quad |R_2|\le C\,\|\delta y\,\delta u\|^2_{L_2(\Omega)}\ .$$

Then

$$(26)\quad |R_1|+|R_2|=o\,(\|\delta u\|_{W^{0,1}_\infty(\Omega)}).$$

By substituting (26) in (21), we obtain

$$(27)\quad J(u+\theta\,\delta u)-J(u)=-\int_\Omega \mathrm{Re}(y\overline{\Phi})\,(\theta\,\omega)\,dx\,dt+O\,(\theta).$$

Hence, in light of the variation functional, we have

$$(28)\quad \delta J(u,\omega)=\lim_{\theta\to 0}\frac{J(u+\theta\ u)-J(u)}{\theta}=-\int_\Omega \mathrm{Re}(y\overline{\Phi})\,\omega\,dx\,d$$

and this proves the differentiability of the functional and gradient formulae of the function $J(u)$. This completes the proof of the theorem.

Using Tikhinov method [7], we define the following functional

$$(29)\quad J_m(u)=J(u)+\alpha^m\int_0^1\int_0^T|\,u(x,t)-\omega(x,t)|^2\,dx\ dt\,.$$

and $\omega(x,t)\in L_2(\Omega)$.

4. Discrete Problem

We consider the set of node values $\{x_j, t_k\},\ x_j = x_0 + J\,h,\ j=\overline{0,M}$

$t_k = t_0 + k\,\tau,\ k=\overline{0,N},\ M=\frac{l}{h},\ N=\frac{T}{\tau}$ and the following notations [8]:

$$(30)\qquad (y_j^k)_{\bar{x}} = \frac{y_j^k - y_{j-1}^k}{h},\ (y_j^k)_x = \frac{y_{j+1}^k - y_j^k}{h},$$

$$(31)\qquad (y_j^k)_{\bar{t}} = \frac{y_j^k - y_j^{k-1}}{\tau},\ (y_j^k)_{\bar{x}x} = \frac{y_{j+1}^k - 2y_j^k + y_{j-1}^k}{h}$$

After applying the numerical integration formula [8],we have the discertisation of the optimal control problem (1)-(5) as follows: Let it is required to minimize the functional

$$(32)\qquad I_m([u]) = \tau \sum_{k=0}^{N} \{\alpha_0 [y_0^k - f_0^k]^2 - \alpha_1 [y_M^k - f_1^k]^2\}$$

$$+\nu^m \tau \sum_{k=1}^{N} \{h \sum_{j=1}^{M-1} \left|u_0^k - \omega_0^k\right|^2 + \frac{1}{2}\left|u_M^k - \omega_M^k\right|^2 + \frac{1}{2}\left|u_j^k - \omega_j^k\right|^2\}$$

on the control set

$$U_N^M = \left\{ \begin{array}{l} [u] : [u] = (u_j^k),\ \alpha_0 \le u_j^k \le \alpha_1,\ j=\overline{0,M},\ k=\overline{0,N}, \\ \left|(u_j^k)_t\right| \le \alpha_2,\ j=\overline{0,M},\ k=\overline{2,N} \end{array} \right\}$$

under the conditions

$$(33)\quad i\,(y_j^k)_{\bar{t}} + B_0\,(y_j^k)_{\bar{x}x} - u_j^k\, y_j^k = f_j^k,\ j=\overline{1,M-1},\ k=\overline{1,N}$$

$$(34)\qquad y_j^0 = 0,\quad j=\overline{0,M},$$

$$(35)\quad \frac{2B_0}{h}(y_0^k)_x = f_0^k - i\,[(u_0^k)_{\bar{t}} - u_0^k\, y_0^k],\ k=\overline{1,N}$$

$$(36)\quad -\frac{2B_0}{h}(y_M^k)_x = f_M^k - i\,[(u_M^k)_{\bar{t}} - u_M^k\, y_M^k],\ k=\overline{1,N}$$

Now, the discrete gradient formulae will be given as follows:

Theorem 2

The functional $J_m(u)$ is differentiable, and its gradient satisfies

$$(37)\quad (I_m^{/}([u]))_j^k = -\mathrm{Re}\,(y_j^k\,\Phi_j^k) + 2\,\alpha^m\,(u_j^k - \omega_j^k),$$

where $j=\overline{0,M-1},\ k=\overline{1,N}$ and Φ_j^k is the solution of discrete adjoint problem:

$$(38)\quad i\,(\Phi_j^k)_{\bar{t}} + B_0\,(\Phi_j^k)_{\bar{x}x} - u_j^k\,\Phi_j^k = 0,\ j=\overline{1,M-1},\ k=\overline{1,N-1}$$

$$(39)\qquad \Phi_j^N = 0,\quad j=\overline{0,M},$$

$$(40)\quad (\Phi_j^k)_x + \frac{2\alpha_0}{B_0}[y_0^k - f_0^k] = \frac{h}{B_0}[u_0^k\,\Phi_0^k - i\,(\Phi_0^k)_t],\ k=\overline{1,N-1}$$

$$(41)\quad (\Phi^k_j)_{\bar{x}} + \frac{2\alpha_0}{B_0}[y^k_M - f^k_1] = \frac{h}{B_0}[u^k_M \Phi^k_M - i\,(\Phi^k_M)_t], k=\overline{1,N-1}$$

5. Solution of Control Problem

5.1 The Projection Gradient Method

Here we describe the projection gradient method [9] for the solution of the optimal control problem such as: construct a sequence $u_{n+1\,m}$ by setting

$$(42)\quad [u]_{n+1m} = P_{U^M_N}\left\{[u]_{nm} - \nu_n \left(I^1_m([u]_{nm})\right)\right\}$$

where $P_{U^M_N}(u)$ is the project on the set U^M_N . In the first we define $(\overset{-k}{u_j})_{nm}$ in the form

$$(43)\quad (u^k_j)_{n+1m} = \begin{cases} \Psi_1 & \alpha_0 \le \Psi_2 \le \alpha_1 \\ \alpha_0 & \Psi_2 < \alpha_0 \\ \alpha_1 & \Psi_2 > \alpha_1 \end{cases}$$

where

$$(44)\quad \Psi_2 = [u^k_j]_{nm} - \nu_n \left(I^1_m([u]_{nm})\right)^k_j$$

$$(45)\quad \Psi_1 = [u^k_j]_{nm} + \nu_n \left(I^1_m([u]_{nm})\right)^k_j$$

and $j=\overline{0,M}$, $k = \overline{1,N}$, $n=0,1,...$, $m = 0,1,...$

Using the above sequence we construct the project in the form

$$(46)\quad (u^1_j)_{n+1m} = (\overset{-1}{u_j})_{n+1m}$$

$$(47)\quad (u^k_j)_{n+1m} = \begin{cases} \Theta_0 & \Theta_1 \le \Psi_2 \le \Theta_2 \\ \Theta_1 & (u^k_j)_{nm} < \Theta_1 \\ \Theta_2 & (u^k_j)_{nm} > \Theta_2 \end{cases}$$

where

$$\Theta_0 = (\overset{-k}{u_j})_{n+1m} + \nu_n \left(I^1_m([u]_{nm})\right)^k_j$$

$$\Theta_1 = -\tau\,\alpha_2 + (\overset{-k-1}{u_j})_{n+1m}\ ,\ \Theta_2 = \tau\,\alpha_2 + (\overset{-k-1}{u_j})_{n+1m}$$

$j=\overline{0,M}$, $k = \overline{2,N}$, $n=0,1,...$, $m = 0,1,...$

5.2 Numerical Algorithm

With the gradient obtained, the following gradient type algorithm can then be developed for the optimal value of $u*$ based on the projection gradient method (PGM)which described in the above section.

The outlined of the algorithm for solving control problem are as follows:

Step 1: Choose an initial control $u^{(n)} \in U,\ n=0$.

If $I^{/}(u^{(n)})=0$, $u^{(n)}$ is the solution of the problem.

Step 2 : At each iteration n do

Solve the state problem, then find $y(.,u^{(n)})$.

Solve the adjoint problem for (1)-(3), then find

$\Phi(.,u^{(n)})$. Find optimal control $u_*^{(n+1)}$ using PGM.

End do.

Step 3: Test the optimality of $u^{(n+1)}$.

If $u^{(n+1)}$ is optimum, stop the process.

Otherwise, go to Step 4.

Step 4 Set $u^{(n+1)} = u^{(n)}$, $n = n + 1$ and go to Step 2.

6. Numerical Results

Designed algorithm is implemented as a FORTRAN routine [10]. Numerical experiment is carried out to check its performance. The initial data of the problem (1)-(5) are taken as follows:

$$\alpha_0 = \alpha_1 = \alpha_2 = l = T = 1,\ \varepsilon = 0.5E-03$$

$$f_0 = it,\ f_1 = i(1+t),\ \varphi(x) = i\,x,\ u^{(0)} = 1.0$$

$$\omega(x,t) = 1 + \frac{x+t}{2},\ f(x,t) = -1 - i(x+t)(x^2+t+1)$$

The number of division of the intervals was taken as $N = M = 20$. The computed control values of u_j^{13}, $j = \overline{0,N}$ the values of relative error are shown in Tables 1 ,2 and the 3D plots of the optimal control and initial values are presented in Figures 1,2 . The optimal value of the cost functional is $J_* = \inf_{u \in U} J(u) = J(u_*) = 0.48526\,E-03$.

The computed control values of u_j^{13}, $j = \overline{0,N}$			
0.15592E+01	0.15950E+01	0.16301E+01	0.16641E+01
0.17221E+01	0.17464E+01	0.17714E+01	0.18021E+01
0.18332E+01	0.18602E+01	0.19112E+01	0.19830E+01
0.20679E+01	0.21474E+01	0.22155E+01	0.22837E+01
0.23625E+01	0.24368E+01	0.24078E+01	0.24324E+01
0.24718E+01			

The values of relative error of u_j^{13}, $j = \overline{0,N}$			
0.025528	0.004683	0.012478	0.02563
0.050038	0.030459	0.048194	0.046236
0.041988	0.032024	0.033075	0.042328
0.055069	0.061758	0.060028	0.056052
0.054668	0.049211	0.000924	0.028029
0.049297			

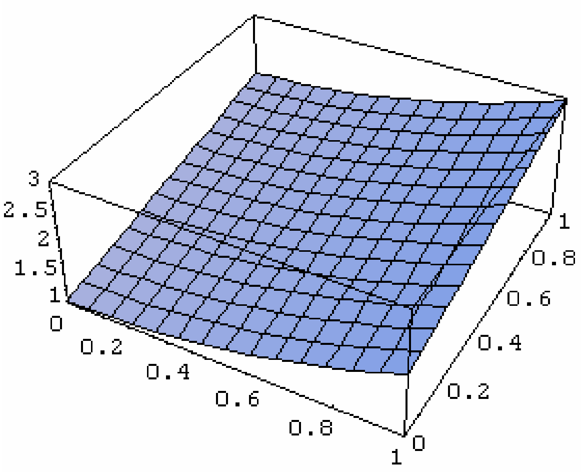


Fig. 1. Optimal control $u_*(x,t)$

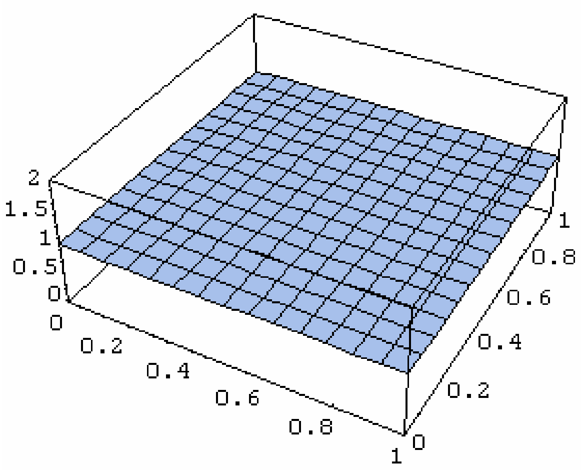


Fig. 2. Initial control $u_0(x,t)$

REFERENCES

[1] A. J. Kearsley, The use of optimization techniques in the solution of partial differential equations from science and engineering, PhD, Housten, Texas,1996.

[2] K. Kime, Finite difference approximation of control via the potential in a 1-D Schrodinger equation, Journal of Differential equations, 2000(26),1--10(2000).

[3] N. U. Ahmed and K. L. Teo, Optimal control of distributed parameter systems, North Holland New York, Oxford(1981).

[4] S. H. Farag and M. H. Farag, Necessary optimality conditions in control problems for hyperbolic equations, Journal Egypt. Math. Soc., 8(2),1--10(2000).

[5] O. A. Ladyzhenskaya, Boundary value problems of mathematical physics ,Nauka, Moscow, Russian(1973).

[6] V. P. Mikhailov, Partial differential equations,Nauka, Moscow, Russian(1983).

[7] A. N. Tikhonov and N. Ya. Arsenin, Methods for the solution of incorrectly posed problems, Nauka, Moscow, Russian(1974).

[8] A. H. Khater, A. B. Shamardan, M. H. Farag and A. H. Abel-Hamid, Analytical and numerical solutions of a quasilinear parabolic optimal control problem, Journal Comput. and Appl. Math., 95,29--43(1998).

[9] J. T. Betts, Survey of numerical methods for trajectory optimization, Journal of Guidance control and dynamics, 21(2),193--207(1998).

[10] D. Kraft, Algorithm 773:TOMP-Fortran modules for optimal control calculations, ACM Transactions of mathematical software, 20(2),262--281(1994).

Author Information

M. H. FARAG : Department of Mathematics and Computer Science; Faculty of Education, Ibri, Sultanate of Oman. or: Faculty of Science ,El-Minia University, Egypt. E-mail: farag5358@yahoo.com

AUTOMATIC TRANSLATION OF MSC DIAGRAMS INTO PETRI NETS

S. Kryvyy, L. Matvyeyeva, M. Lopatina

Abstract: Development-engineers use in their work languages intended for software or hardware systems design, and test engineers utilize languages effective in verification, analysis of the systems properties and testing. Automatic interfaces between languages of these kinds are necessary in order to avoid ambiguous understanding of specification of models of the systems and inconsistencies in the initial requirements for the systems development.

Algorithm of automatic translation of MSC (Message Sequence Chart) diagrams compliant with MSC'2000 standard into Petri Nets is suggested in this paper. Each input MSC diagram is translated into Petri Net (PN), obtained PNs are sequentially composed in order to synthesize a whole system in one final combined PN. The principle of such composition is defined through the basic element of MSC language — conditions. While translating reference table is developed for maintenance of consistent coordination between the input system's descriptions in MSC language and in PN format. This table is necessary to present the results of analysis and verification on PN in suitable for the development-engineer format of MSC diagrams. The proof of algorithm correctness is based on the use of process algebra ACP. The most significant feature of the given algorithm is the way of handling of conditions. The direction for future work is the development of integral, partially or completely automated technological process, which will allow designing system, testing and verifying its various properties in the one frame.

Keywords: MSC diagram, MSC language, condition, automatic translation, Petri Net.

Introduction

While designing and developing of either software or hardware it is of vital importance to detect and remove defects in product on its early stages in order to avoid time and resource losses. Development-engineers and test engineers (verifiers) use in their work different approaches and specification languages, that eventually leads to ambiguous understanding of the same portion of a project, to inaccuracies, incompletenesses or even to the inconsistencies in the initial requirements for the development. Development-engineers usually utilize languages intended for design purposes (as VHDL, MSC, SDL, UML and so on), while test engineers (verifiers) utilize languages effective in verification and testing (languages of mathematical logics, automata theory, algebraic and net languages). The way-out of this situation is a development of automatic interfaces between languages of these kinds. Given work is devoted to the development of automatic interface between the languages MSC (Message Sequence Chart) and PN (Petri Nets). The work suggests an algorithm of automatic translation of MSC diagrams compliant with MSC'2000 language standard [ITU-TS, 2000] into Petri Nets, which allow verifying automatically a lot of properties of the system under design. The algorithm works on a certain subset of MSC'2000 language.

1. Syntactic MSC constructions

MSC is a modelling technique that uses a graphical interface, which was standardized by ITU (International Telecommunication Union, earlier CCITT). It is usually applied to applications of the telecommunication domain, since they have properties of distributed reactive real-time systems, often in combination with SDL language [Grabowski, 1991]. These very properties of the systems make an MSC with possibility of scenario describing extremely suitable as for specification so for testing purposes. This means that MSC can be applied on every stage of system development, even on the stage of test case development. MSC describes message flow between the instances, which present asynchronously communicating objects of the system or system entities like blocks, services or processes of the system. One MSC diagram describes a certain portion of system behaviour or a scenario of communication between the instances.

MSC has two syntactical representations: textual and graphical, which are in one-to-one relation according to a standard. Basic elements of the language are those which define message flow, namely, *instance*, *message*, *action*, *set*->*reset*, *set*->*time-out*, *stop*, *create* and *condition*. An example of an MSC is presented on Figure 1(a). As far as this example is of illustrative kind, it only introduces a minimal set of possible MSC constructions as instances and messages. Let's describe basic elements of the language MSC'2000, which are considered in the given work.

1.1. Instances, messages and system environment.

Instance is a basic primitive of MSC, which in graphics is presented as vertical line with its name.

Message transmissions, which are acts of communication between instances, are presented by horizontal arrows with possible curve or tilt under angle for reflecting "overtaking" or "intersection" of messages. The beginning of the vector marks a sending of the message and its ending marks receiving of the message. Evens of sending and receiving of the messages are ordered along the instances so that sending of the message always happens earlier than its receiving. There is one more rule in standard MSC'2000 for ordering events along the instances: everything located above happens earlier than that located below. A MSC diagram imposes a partial ordering on the set of events being contained. A binary relation which is transitive, antisymmetric and reflexive is called partial order. The partial ordering can be described by its connectivity graph.

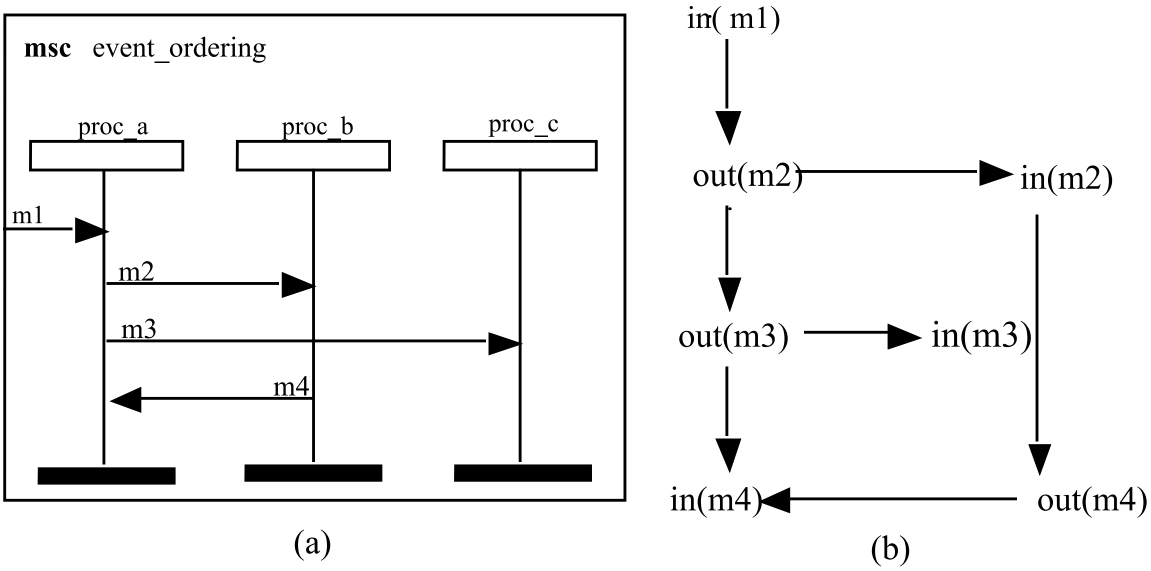


Figure 1

At the Picture 1(b) there is a graph, which reflects the order of events on instances in the diagram "msc event_ordering" (Figure 1(a)). Event out(m1) means sending of the message m1, in(m1) – receiving of message m1.

Environment of the system (the set of instances) is presented by the borders of MSC.

1.2. Conditions

Condition is used as for restricting or defining a set of MSC traces through indicating states of the system so for defining the composition of one MSC diagram from the several MSC diagrams. Condition can describe a global state of the system which is extended to all MSC instances existing in the time of this condition; it also can describe a state for a certain subset of MSC instances. In the first case condition is called global. Instances presenting dynamic objects can be began and finished, so far globality of a state considers dynamically changing set of instances.

Standard MSC'2000 [ITU-TS, 2000] defines conditions of two types: setting condition and guarding condition. Conditions of the first type are those which describe the current state of the system. Conditions of the second type restrict behaviour of the MSC to execution of events in a certain part of MSC depending on the value of the given guarding condition.

Besides of composition role, conditions according to the standard [ITU-TS, 2000] are the means of events synchronization. For example, if two instances share one and the same condition, then for each message between these instance its sending and receiving events shall happen both before or both after setting of the condition. If two conditions are ordered directly sharing the common instance, or indirectly through conditions on other instances, then this order must be respected on all instances that share these two conditions.

2. MSC semantics

The first version of MSC language standard defined the semantics incompletely and informally, however in ideal development of the language and its semantics shall go in parallel. Need of semantics standardization was becoming apparent, as even MSC experts could not always agreed on interpretation of the particular properties. Associated with this situation and extension of MSC language application in 1992 three new approaches of MSC semantics defining were submitted to standardization committee CCITT (now ITU-TS or Telecommunication Standardization section of the International Telecommunication Union).

The first approach was based on the theory of finite automata.

The second one was based on theory of Petri Nets using partially ordered sequences of events in the system. Given semantics is known to be extremely suitable for modelling of distributed asynchronous systems.

The third approach was based on process algebra ACP (Algebra of Communicating Processes) and interleaving model, when system is modelled by the sequence of transitions supposing that events are atomic and has no duration, and in every moment of time only one event can be executed. Semantics of interleaving simulates independence (asynchronism) among the subsystems (instances) through nondeterministic interleaving of independent parallel activities.

Every of the three approaches given for definition of MSC language semantics has its advantages and disadvantages, but the committee for standardization chose third approach to be the basis for formal definition of MSC language semantics. MSC language semantics based on process algebra [Bergstra, 1984] was defined for textual representation of MSC diagrams [ITU-TS, 1995] in expressions of process algebra that was called denotation semantics. Operational semantics is defined through addition of transitional rules to algebraic expressions. So, operational semantics is reflection of MSC-specifications into transition system. Yet it should note that operational semantics of MSC is not defined and standardized formally.

3. Algorithm of translation of MSC diagrams into Petri Net

The first progresses in defining MSC semantics basing on Petri Nets were presented in [Grabowski, 1993]. Nowadays research work on translating of MSCs into Petri Nets goes on and its results are covered, for example, in [Kluge, 2000], [Heymer, 2000], [Kluge, 2001]. However the authors develop semantics for the MSC'96 and MSC'2000 standards basing on Petri Nets, and, what's more, elaborate on MSCs into the Petri Nets translation, an automatic translation is not considered in these papers.

3.1. Description of algorithm

The following is the formalized description of the translation algorithm.

INPUT: A set of the MSC diagrams in the basic subset of the language MSC'2000.

OUTPUT: A Petri Net adequate to the set of initial MSC diagrams.

METHOD: Translation of every MSC diagram of the input set into a Petri Net is performed in two stages in parallel with composing of corresponding to each MSC diagram Petri Nets into one final combined Petri Net (synthesis).

Begin
Stage 1. Building of the partial-order graph to reflect events order in the initial MSC diagrams imposed by static requirements of MSC'2000 standard [ITU-TS, 2000] (see Figure 1).
Stage 2. Translating of the partial-order graph obtained at the stage 1 into the Petri Net.
End

Let's give more detailed description of the algorithm's stages and start with the description of how synthesis is proceeds.

Detailed description of the synthesis

Each input MSC diagram is being translated into PN, the PNs are sequentially “glued” in order to compose (synthesize) a whole system in one PN, while the principle of such composition is defined at the level of MSC language - through *conditions*. Let's detail semantics of the basic element of MSC language *condition* as it was described in the MSC'2000 standard.

According to the MSC'2000 standard *condition* defines a system state of instance, which it covers . A system state of the instance is interpreted not as a current global state of the whole system, but as a set of the current values of a certain subset of attributes coherent with the instance (system object/entity), another words, *condition* is a precondition of occurring the event in the system. Let's explain this statement. MSC language was created to specify systems with local interconnections among the asynchronous parallel processes. Asynchronous models are typically built on cause-and-effect relation among events rather than on clocked sequence of system state changes. Asynchronous systems also present time moments or intervals as events. So event in this model is considered to be either atomic or compound with internal structure formed from “sub-events”. Thus, *condition* is a precondition of occurring the event in the system and simultaneously a synchronizing action. The steps of the synthesis are carried out according to the following rules, on the assumption of fulfilling the following requirements.

The requirements:

1. We only consider a subset of MSC'2000, including the elements: instances, message inputs and outputs, setting conditions in textual representation. It is supposed, that all input diagrams are syntactically correct and satisfy the static requirements of the MSC'2000 standard. For example, for each event of a message sending the diagram shall have a pair event — a message consumption.

2. Naming of instances is complete and exact.

3. Naming of conditions is complete and exact, meaning of the conditions do not influence synthesis in any way.

4. Each instance shall have initial and final conditions either local or, probably, shared by several instances. The given requirement is not referred to a case of creation and termination of the instance within the scope of the given diagram.

The condition is called *initial*, if the diagram has not any event or condition which precedes it, and *final*, if there is no event or condition after the given condition. Not only singular but also multiple initial and final conditions are possible, in logic they can be represent as conjunction of all initial conditions, and, correspondingly, conjunction of all final conditions.

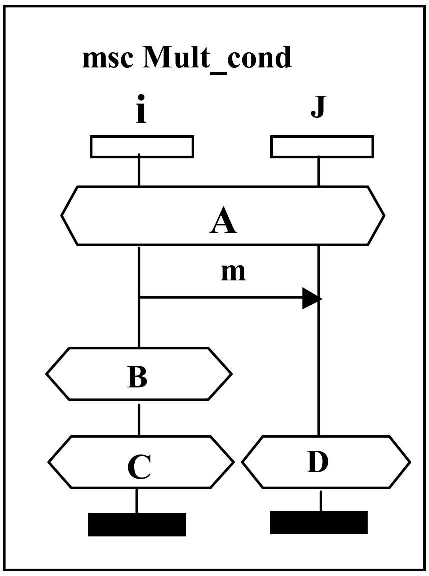


Figure 2

For example, at Figure 2 conditions **B** and **C** represent a multiple final condition, **D** — singular final condition or simply final condition, **A** is an initial condition.

5. Each diagram from the input set shall satisfy the following: every initial and final condition of MSC shall cover a minimum one instance, on which an event occurs (in given case, either sending or consumption of the message).

6. "Gluing" is forced and carried out according to the following rules.

The Rules:

1. If the first MSC diagram has a final condition, which corresponds (has the same name) to the initial condition of the second MSC diagram, and these conditions cover the same set of instances in the both diagrams, than the first and the second diagrams can be “glued” together via the given condition regardless of the fact that this condition is either global or local for the diagrams. For example, at Figure 3 **msc Z** is a synthesis or composition of **msc X** and **msc Y**. The dashed line shows places of "gluing".

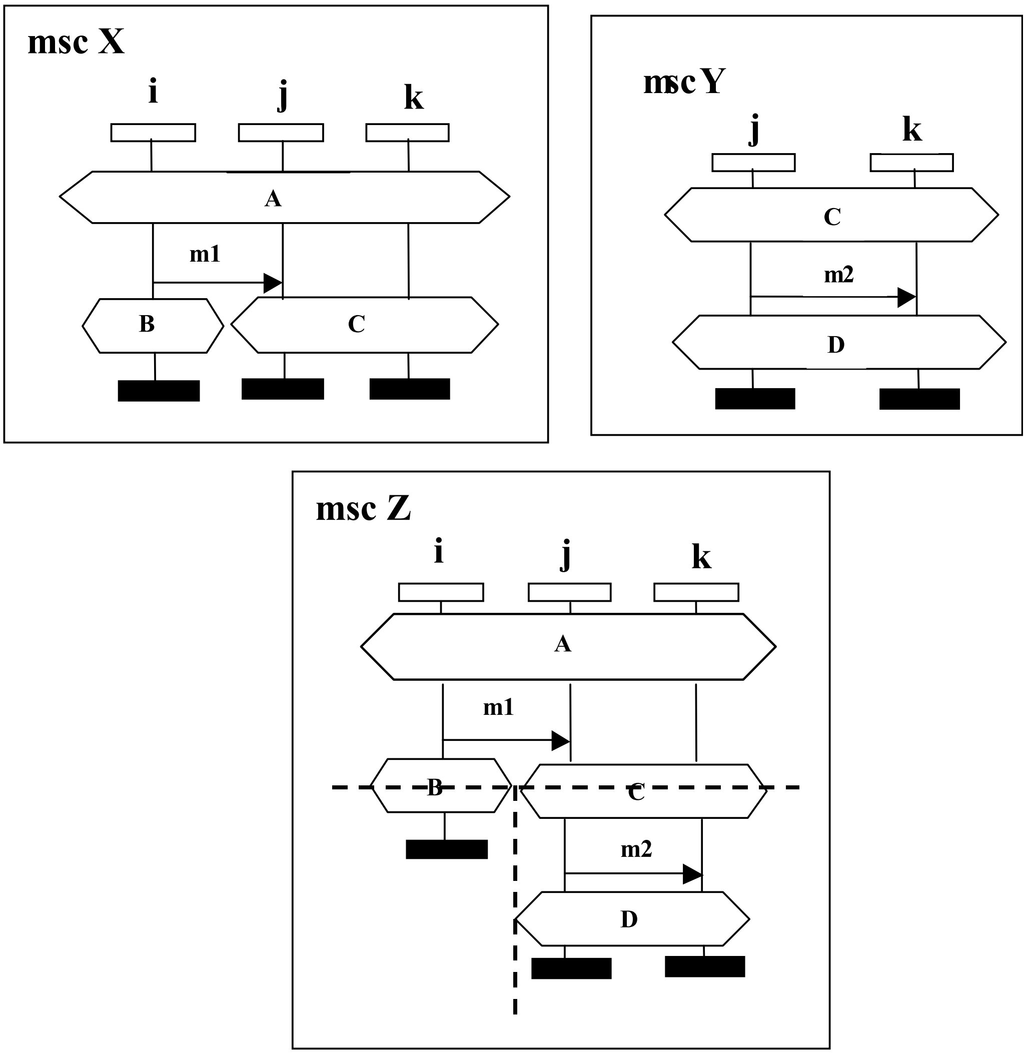


Figure 3

This rule also assumes the opportunity to “glue" by condition not only two diagrams in sequence, but also to “glue" together more than two diagrams at once (“multiple gluing”), this means that "gluing" is a non-deterministic alternative composition in opposite to the delayed choice operator for alternatives. As a result of gluing we receive an MSC diagram presenting a set of possible alternative traces in the system. Let's consider, for example, Figure 4. There are two possible continuations of the diagram **msc X**: it is **msc Y** and **msc Z**. And the choice is made when the condition **C** occurs and it is not delayed until the events presented on **msc Z** (**msc Result 1**) and on **msc Y** (**msc Result 2**) begin to differ from each other, namely, before passing of the messages **m** and **k**.

2. The following is a very important rule for "gluing" of MSC diagrams by **overlapped** and **multiple** conditions:

a) conditions occur simultaneously and are equal to the conjunction of these conditions,

b) the order of their enumeration in the diagrams is of no importance because of simultaneous occurrence of conditions,

c) these conditions can glue together independently from each other.

Let's consider an example at Figure 5. Each of MSC diagrams X1, X2, X3 is a possible continuation of **msc X**. The conditions **B** and **C** are overlapped in **msc X** and **msc X3**.

All requirements and rules mentioned above do not contradict the MSC'2000 standard, and only specify it. Composing of one common Petri Net is performed in accordance with semantics of the synthesis of MSC diagrams described above. A table of initial and final conditions of the initial MSCs plays a key role in the synthesis, as it contains all information necessary for composing. It is formed on the stage 1 during sequential processing of MSCs. Thus, the gluing is performed along with translation of MSCs applying the table of initial and final conditions.

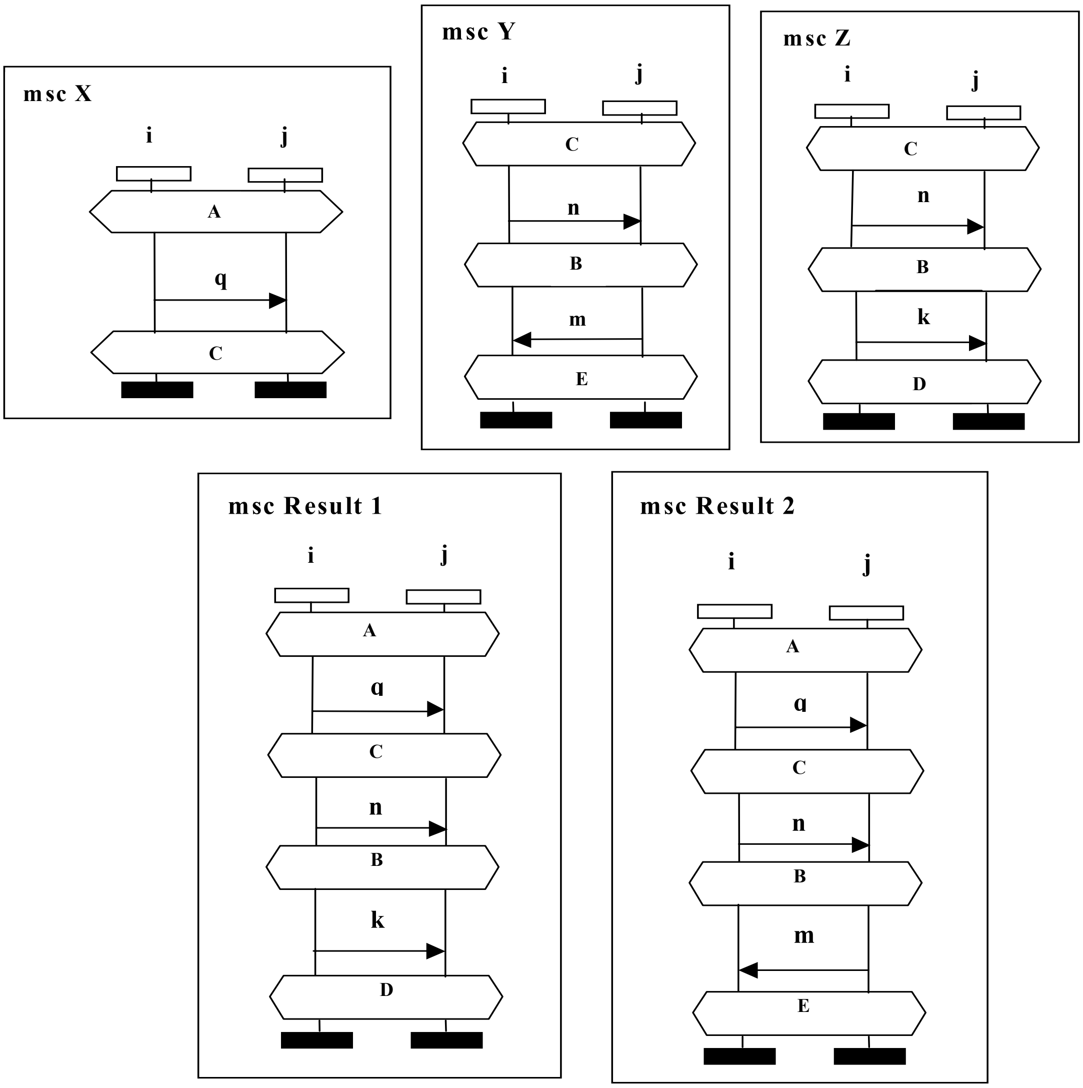


Figure 4

Detailed description of stage 1. Partial-order graph for system events is built according to the following rules:

1. Directed edges of the graph set an order of the events.

2. Events of sending and consumption of messages and conditions correspond to the graph nodes. It is necessary to emphasize, that a condition is also represented by an event, namely, event of synchronization. Moreover the graph defines nodes of **two** types: a graph node which denotes intermediate condition or event of message consumption/sending, and a graph node which denotes initial or final conditions. They differ in the way of translation into the elements of Petri Net during the second stage.

Detailed description of stage 2. Rules for the second stage of translation (translation of the graph into the Net) are the following:

1. Each directed edge of the partial-order graph is translated into a place of Petri Net.

2. An arrow of each directed edge of the graph corresponds to an arrow of Petri Net that directs tokens' flow in Petri Net.

3. A graph node of the **first** type (denoted intermediate conditions and events of an message input (consumption) and message output (sending)) is translated into a transition of Petri Net.

4. A graph node of the **second** type (denoted final and initial conditions) is translated into combination of transition and place of Petri Net. This place is a joint element for "gluing" into the common Petri Net (representing input system as a whole) .

While translating reference table is developed for maintenance of consistent coordination between the input system's descriptions in MSC language and in Petri Net format. This table is necessary to present the results of analysis and verification on Petri Net in suitable for the development-engineer format of MSC diagrams.

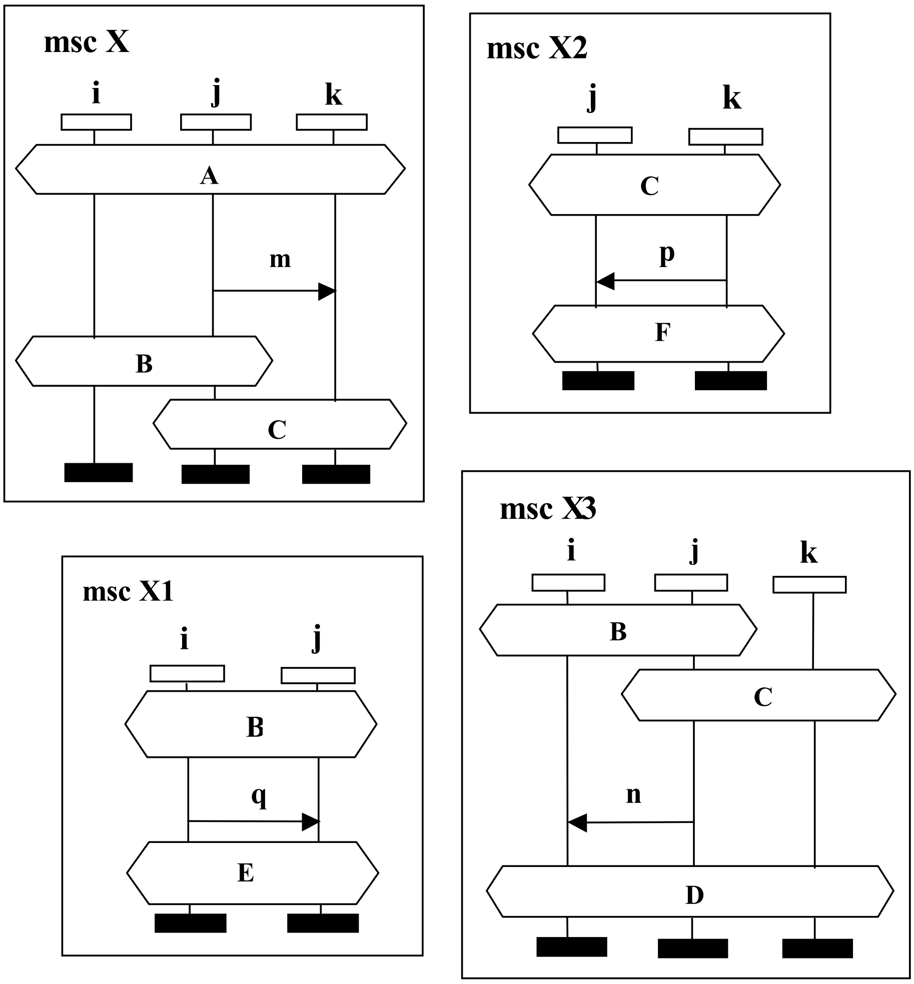


Figure 5

Unfortunately, limited size of the paper does not allow to place here the proof of algorithm correctness even in the reduced version. We note only that the proof of algorithm correctness is based on the use of process algebra ACP.

Conclusion

Summing up, we note, that the algorithm of an automatic translation, presented in the paper, considers only the subset of MSC language, therefore extending of this subset to the maximum or even up to the whole MSC language is an obvious direction of the further research. The most significant feature of the given algorithm is the way of handling of conditions, since the literature indicates this problem in translation process as the most difficult. What is also important is obtaining of necessary experience for developing analog translators for the languages: SDL, UML, etc. The ultimate goal of this research is development of integral, partially or completely automated technological process, which will allow designing system, testing and verifying its various properties in the one frame.

Bibliography

[Grabowski, 1993] J. Grabowski, P. Graubmann, E. Rudolph. Towards a Petri Net Based Semantics Definition for Message Sequence Charts, In O.Fergemand and A.Sarma, editors, SDL'93 Using Objects, Proceedings of the 6th SDL Forum, Darmstadt, 1993. Elsevier Science Publishers B.V.

[Grabowski, 1991] J. Grabowski, P. Graubmann, E. Rudolph. Towards an SDL-Design-Methodology Using Sequence Cart Segment, SDL'91 Evolving Methods - O.Fergemand and A.Sarma, editors, North-Holland, 1991.

[CCITT, 1992]. CCITT Recommendation Z.120: Message Sequence Chart (MSC). Geneva,1992.

[Reniers, 1995] M. A. Reniers.Static semantics of Message Sequence Charts. In SDL'95 with MSC in CASE, Proceedings of the Seventh SDL Forum, Oslo, 1995. Elsevier Science Publishers B.V.

[ITU-TS, 1995] ITU-TS Recommendation Z.120 Annex B: Algebraic semantics of Message Sequence Charts. ITU-TS, Geneva, 1995.

[Котов, 1984] В.Е.Котов Сети Петри. М.:Наука, 1984, 157 стр.

[ITU-TS, 2000] ITU-TS Recommendation Z.120: Message Sequence Chart (MSC). ITU-TS, Geneva, 2000.

[Belina, 1992] F. Belina. SDL Methodology Guidelines, CCITT, Geneva, May 1992.

[Bergstra, 1984] J.A. Bergstra, J.W. Klop. Process Algebra for Synchronous Communication, Inf.&Control 60,pp.109-137,1984.

[Mauw, 1995] S. Mauw, M.A. Reniers. Thoughts on the meaning of conditions. Experts meeting SG10, St.Petersburg TD9016, ITU-TS,1995.

[Kluge, 2000] O. Kluge. Time in Message Sequence Charts Specifications and How to Derive Stochastic Petri Nets. In: Proceedings of the Third International Workshop on Communication Based Systems (CBS3), Berlin, March 2000.

[Heymer, 2000] S. Heymer. A Semantic for MSC based on Petri-Net Components, SIIM Technical Report A-00-12, Informatik-Berichte Humboldt-Universitдt zu Berlin, June 2000.

[Kluge, 2001] O. Kluge, J. Padberg, H. Ehrig. Modeling Train Control Systems: From Message Sequence Charts to Petri Nets, Technische Universitдt Berlin,http://cs.tu-berlin.de/SPP/index.html, 2001

Authors information

Sergiy Kryvyy - Institute of cybernetics, NAS of Ukraine, Pr. Glushkova, 40, Kiev-03187, Ukraine; e-mail: krivoi@i.com.ua

Lyudmila Matvyeyeva - Institute of cybernetics, NAS of Ukraine, Pr. Glushkova, 40, Kiev-03187, Ukraine; e-mail: luda@iss.org.ua

Mariya Lopatina - Institute of cybernetics, NAS of Ukraine, Pr. Glushkova, 40, Kiev-03187, Ukraine; e-mail: marichkay@mail.ru

REPRESENTING REFLECTIVE LOGIC IN MODAL LOGIC

Frank M. Brown

Abstract: The nonmonotonic logic called Reflective Logic is shown to be representable in a monotonic Modal Quantificational Logic whose modal laws are stronger than S5. Specifically, it is proven that a set of sentences of First Order Logic is a fixed-point of the fixed-point equation of Reflective Logic with an initial set of axioms and defaults if and only if the meaning of that set of sentences is logically equivalent to a particular modal functor of the meanings of that initial set of sentences and of the sentences in those defaults. This result is important because the modal representation allows the use of powerful automatic deduction systems for Modal Logic and because unlike the original Reflective Logic, it is easily generalized to the case where quantified variables may be shared across the scope of the components of the defaults thus allowing such defaults to produce quantified consequences. Furthermore, this generalization properly treats such quantifiers since all the laws of First Order Logic hold and since both the Barcan Formula and its converse hold.

Keywords: Reflective Logic, Modal Logic, Nonmonotonic Logic.

1. Introduction

One of the simplest nonmonotonic logics which inherently deals with entailment conditions in addition to possibility conditions in its defaults is the so-called Reflective Logic [Brown 1989]. The basic idea of Reflective Logic is that there are some assumptions Γ and some non-logical "inference rules" of the form:

$$\frac{\alpha : \beta_1,\ldots,\beta_m}{\chi}$$

which suggest that χ may be inferred whenever α is inferable and each $\beta_1,\ldots,\beta_m$ is consistent with everything that is inferable. Such "inference rules" are not recursive and are circular in that the determination as to whether χ_i is derivable depends on whether β_j is consistent which in turn depends on what was derivable from this and other defaults. Thus, tentatively applying such inference rules by checking the consistency of $\beta_1,\ldots,\beta_m$ with only the current set of inferences produces a χ_i result which may later have to be retracted. For this reason valid inferences in a nonmonotonic logic such as Reflective Logic are essentially carried out not in the original nonmonotonic logic, but rather in some (monotonic) metatheory in which that nonmonotonic logic is defined. [Brown 1989] explicated this intuition[2] by defining Reflective Logic in terms of the set theoretic proof theory metalanguage of First Order Logic (i.e. FOL) with the following fixed-point expression:

$$'\kappa=(\mathrm{rl}\ '\kappa\ \{'\Gamma_i\}\ '\alpha_i{:}'\beta_{ij}/'\chi_i)$$

where rl is defined as: $(\mathrm{rl}\ '\kappa\ \{'\Gamma_i\}\ '\alpha_i{:}'\beta_{ij}/'\chi_i) =_{\mathrm{df}} (\mathrm{fol}(\{'\Gamma_i\}\cup\{'\chi_i{:}('\alpha_i\varepsilon'\kappa)\wedge\wedge_{j=1,m_i}('(\neg\beta_{ij})\not\in'\kappa)\}))$

where $'\alpha_i$, β_{ij}, and $'\chi_i$ are the closed sentences of FOL occurring in the ith "inference rule" and $\{'\Gamma_i\}$ is a set of closed sentences of FOL and $'\Gamma_i$ is the ith sentence in that set. A closed sentence is a sentence without any free variables. fol is a function which produces the set of theorems derivable in FOL from the set of sentences to which it is applied. The quotations appended to the front of these Greek letters indicate references in the metalanguage to sentences of the FOL object language. Interpreted doxastically this fixed-point equation states:

> the set of closed sentences which are believed is equal to:
> the set of closed sentences derived in FOL from
> the union of the set of closed sentences: $\{'\Gamma_i\}$,
> and the set of closed sentences of the form $'\chi_i$ such that for each i,
> the closed sentence $'\alpha_i$ is believed and for each j, the closed sentence $'\beta_{ij}$ is believable.

[2] This explication is simpler but less sophisticated in its properties than that of Default Logic [Reiter 1980]. The fixed-points of both logics obey the laws: $'\kappa=(\mathrm{fol}\ '\kappa)$, $'\kappa\supseteq\{'\Gamma_i\}$, and $((\alpha_i\varepsilon'\kappa)\wedge\wedge_{j=1,m_i}('(\neg\beta_{ij})\not\in'\kappa))\rightarrow('\chi_i\varepsilon'\kappa)$. However, the fixed points of Default Logic are a subset of the fixed-points of Reflective Logic, but the converse is in general not true. Moreover, the fixed-points of Reflective Logic are the kernels of the fixed points of Autoepistemic Logic [Moore 1985].

The purpose of this paper is to show that all this metatheoretic machinery including the formalized syntax of FOL, the proof theory of FOL, the axioms of a strong set theory, and the set theoretic fixed-point equation is not needed and that the essence of Reflective Logic is representable as a necessary equivalence in a simple (monotonic) Modal Quantificational Logic. Interpreted as a doxastic logic this necessary equivalence states:

> that which is believed is logically equivalent to
> for each i , Γ_i and for each i, if α_i is believed and for each j, β_{ij} is believable then χ_i

thereby eliminating all mention of any metatheoretic machinery.

The remainder of this paper proves that this modal representation is equivalent to Reflective Logic. Section 2 describes a formalized syntax for a FOL object language. Section 3 describes the part of the proof theory of FOL needed herein (i.e. theorems FOL1-FOL4). Section 4 describes the Intensional Semantics of FOL which includes laws giving the meaning of FOL sentences: M0-M7, theorems giving the meaning of sets of FOL sentences: MS1, MS2, MS3, and laws specifying the relationship of meaning and modality to the proof theory of FOL (i.e. the laws R0, A1, A2, and A3 and the theorems: C1, C2, C3, and C4). The modal version of Reflective Logic, called RL, is defined in section 5 and explicated with theorems MR1-MR6 and SS1-SS2. In section 6, this modal version is shown by theorems RL1 and RL2 to be equivalent to the set theoretic fixed-point equation for Reflective Logic. Figure 1 outlines the relationship of all these theorems in producing the final theorems RL2, FOL4, and MR6.

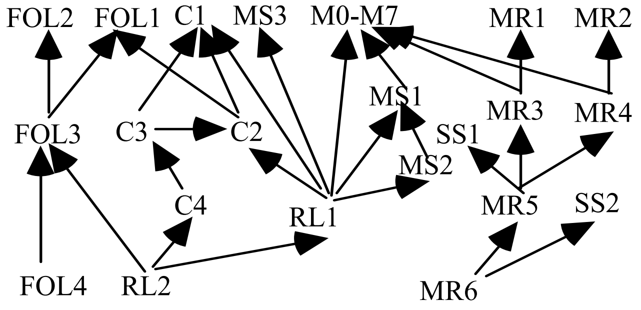


Figure 1: Dependencies among the Theorems

2. Formal Syntax of First Order Logic

We use a First Order Logic (i.e. FOL) defined as the six tuple: ($\rightarrow$, #f, $\forall$, *vars*, *predicates*, *functions*) where $\rightarrow$, #f, and $\forall$ are logical symbols, *vars* is a set of variable symbols, *predicates* is a set of predicate symbols each of which has an implicit arity specifying the number of associated terms, and *functions* is a set of function symbols each of which has an implicit arity specifying the number of associated terms. The sets of logical symbols, variables, predicate symbols, and function symbols are pairwise disjoint. Lower case Roman letters possibly indexed with digits are used as variables. Greek letters possibly indexed with digits are used as syntactic metavariables. γ, γ_1,...γ_n, range over the variables, ξ, ξ_1...ξ_n range over sequences of variables of an appropriate arity, π,π_1...π_n range over the predicate symbols, ϕ, ϕ_1...ϕ_n range over function symbols, δ, δ_1...δ_n, σ range over terms, and α, α_1...α_n, β, β_1...β_n,χ, χ_1...χ_n, Γ_1,...Γ_n,φ range over sentences. The terms are of the forms γ and (ϕ δ1...δn), and the sentences are of the forms ($\alpha \rightarrow \beta$), #f, ($\forall\gamma$ α), and (π δ_1...δ_n). A nullary predicate π or function ϕ is written as a sentence or a term without parentheses. $\varphi\{\pi/\lambda\xi\alpha\}$ represents the replacement of all occurrences of π in φ by $\lambda\xi\alpha$ followed by lambda conversion. The primitive symbols are shown in Figure 2 with their intuitive interpretations.

Symbol	Meaning
$\alpha\rightarrow\beta$	if α then β.
#f	falsity
$\forall\gamma\ \alpha$	for all γ, α.

Figure 2: Primitive Symbols of First Order Logic

The defined symbols are listed in Figure 3 with their definitions and intuitive interpretations.

Symbol	Definition	Meaning		Symbol	Definition	Meaning
$\neg\alpha$	$\alpha \rightarrow \#f$	not α		$\alpha\wedge\beta$	$\neg(\alpha \rightarrow \neg\beta)$	α and β
#t	$\neg$ #f	truth		$\alpha\leftrightarrow \beta$	$(\alpha\rightarrow \beta) \wedge (\beta\rightarrow \alpha)$	α if and only if β
$\alpha\vee\beta$	$(\neg \alpha)\rightarrow \beta$	α or β		$\exists\gamma\ \alpha$	$\neg\forall\gamma\ \neg\alpha$	for some γ , α

Figure 3: Defined Symbols of First Order Logic

The FOL object language expressions are referred in the metalanguage (which also includes a FOL syntax) by inserting a quote sign in front of the object language entity thereby making a structural descriptive name of that entity. In addition to referring to object language sentences, the formalized metalanguage also needs to refer to sets of sentences of FOL. Generally, a set of sentences is represented as: $\{'\Gamma_i\}$ which is defined as: $\{'\Gamma_i$: #t$\}$ which in turn is defined as: $\{s: \exists i(s='\Gamma_i)\}$ where i ranges over some range of numbers (which may be finite or non-infinite). With a slight abuse of notation we also write $'\kappa$, $'\Gamma$ to refer to such sets.

3. Proof Theory of First Order Logic

First Order Logic (i.e. FOL) is axiomatized with a recursively enumerable set of theorems as the set of axioms is itself recursively enumerable and its inference rules are recursive. The axioms and inference rules of FOL [Mendelson 1964] are those given in Figure 4. They form a standard set of axioms and inference rules for FOL.

MA1: $\alpha \rightarrow (\beta\rightarrow \alpha)$ MR1: from α and $(\alpha\rightarrow \beta)$ infer β

MA2: $(\alpha\rightarrow (\beta\rightarrow \rho)) \rightarrow ((\alpha\rightarrow \beta)\rightarrow (\alpha\rightarrow \rho))$ MR2: from α infer $(\forall\gamma\ \alpha)$

MA3: $((\neg \alpha)\rightarrow (\neg \beta))\rightarrow (((\neg \alpha)\rightarrow \beta)\rightarrow\alpha)$

MA4: $(\forall\gamma\ \alpha)\rightarrow \beta$ where β is the result of substituting an expression (which is free for the free positions of γ in α) for all the free occurrences of γ in α.

MA5: $((\forall\gamma(\alpha \rightarrow \beta)) \rightarrow (\alpha\rightarrow(\forall\gamma\ \beta)))$ where γ does not occur in α.

Figure 4: Inferences Rules and Axioms of FOL

In order to talk about sets of sentences we include in the metatheory set theory symbolism as developed along the lines of [Quine 1969]. This set theory includes the symbols ε, $\not\in$, $\supseteq$, $=$, $\cup$ as is defined therein.
The derivation operation (i.e. fol) of any First Order Logic obeys the Inclusion (i.e. FOL1) and Idempotence (i.e. FOL2) properties:

FOL1: (fol $'\kappa)\supseteq'\kappa$ Inclusion

FOL2: (fol $'\kappa)\supseteq$(fol(fol $'\kappa$)) Idempotence

From these two properties we prove:

FOL3: (rl $'\kappa\ '\Gamma\ '\alpha_i$:$'\beta_{ij}/'\chi_i$))=(fol(rl $'\kappa\ '\Gamma\ '\alpha_i$:$'\beta_{ij}/'\chi_i$))

proof: FOL1 and FOL2 imply that (fol(fol $'\kappa$))=(fol $'\kappa$). Since rl begins with fol this implies: $'\kappa$=(fol(rl $'\kappa$)) QED.

FOL4: ($'\kappa$=(rl $'\kappa\ '\Gamma\ '\alpha_i$:$'\beta_{ij}/'\chi_i$))$\rightarrow$($'\kappa$=(fol $'\kappa$))

proof: From the hypothesis and FOL3: $'\kappa$=(fol(rl $'\kappa\ '\Gamma\ '\alpha_i$:$'\beta_{ij}/'\chi_i$)) is derived. Using the hypothesis to replace (rl $'\kappa\ '\Gamma\ '\alpha_i$:$'\beta_{ij}/'\chi_i$) by $'\kappa$ in this result gives: $'\kappa$=(fol $'\kappa$). QED.

4. Intensional Semantics of FOL

The meaning (i.e. mg) [Brown 1978, Boyer&Moore 1981] or rather disquotation of a sentence of First Order Logic (i.e. FOL) is defined to satisfy the laws given in Figure 5 below[3]. mg is defined in terms of mgs which maps each FOL object language sentence and an association list into a meaning. Likewise, mgn maps a FOL object language term and an association list into a meaning. An association list is simply a list of pairs consisting of an object language variable and the meaning to which it is bound.

[3] The laws M0-M7 are analogous to Tarski's definition of truth except that finite association lists are used to bind variables to values rather than infinite sequences. M4 is different since mg is interpreted as being meaning rather than truth.

M0: (mg 'α) =df (mgs '($\forall\gamma$1...γn α)'()) where 'γ1...'γn are all the free variables in 'α
M1: (mgs '($\alpha \rightarrow \beta$)a) $\leftrightarrow$ ((mgs 'α a)$\rightarrow$(mgs 'β a))
M2: (mgs '#f a) $\leftrightarrow$ #f
M3: (mgs '($\forall$ γ α)a) $\leftrightarrow$ $\forall$x(mgs 'α(cons(cons 'γ x)a))
M4: (mgs '(π δ_1...δ_n)a) $\leftrightarrow$ (π(mgn 'δ_1 a)...(mgn 'δ_n a)) for each predicate symbol 'π.
M5: (mgn '(ϕ δ_1...δ_n)a) = (ϕ(mgn 'δ_1 a)...(mgn 'δ_n a)) for each function symbol 'ϕ.
M6: (mgn 'γ a) = (cdr(assoc 'γ a))
M7: (assoc v L) = (if(eq? v(car(car L))) (car L) (assoc v(cdr L)))
where: cons, car, cdr, eq?, if are axiomatized as they are axiomatized in Scheme.

Figure 5: The Meaning of FOL Sentences

For example, the meaning of the sentence "Everything is less than something" is the proposition that everything is less than something. Thus the meaning operator disquotes its argument. Here is an example derivation:

(mg '($\forall$x$\exists$y(< x y)))

Replacing the defined symbols of the object language by primitive symbols of the object language gives:

(mg '($\forall$x(($\forall$y((< x y)$\rightarrow$#f))$\rightarrow$#f))). By M0 this is equivalent to: (mgs '($\forall$x(($\forall$y((< x y)$\rightarrow$#f))$\rightarrow$#f)) '())

By M3 this is equivalent to: $\forall$x(mgs '(($\forall$y((< x y)$\rightarrow$#f))$\rightarrow$#f) (cons(cons 'x x)'()))

By M1 this is equivalent to: $\forall$x((mgs '($\forall$y((< x y)$\rightarrow$#f)) (cons(cons 'x x)'())) $\rightarrow$(mgs '#f (cons(cons 'x x)'())))

By M2 this is equivalent to: $\forall$x((mgs '($\forall$y((< x y)$\rightarrow$#f)) (cons(cons 'x x)'()))$\rightarrow$#f)

We would now like to apply M3 to: (mgs '($\forall$y((< x y)$\rightarrow$#f)) (cons(cons 'x x)'()))

but we cannot since the bound variable x in M3 would capture the variable x which is free in this expression. In order to apply M3 we must first rename the bound variable x in M3 to be some other variable which will not capture any free variables in this expression. In this case we rename the bound x in M3 to be y, and then use that version of M3 to produce the equivalent expression:

$\forall$x(($\forall$y(mgs '((< x y)$\rightarrow$#f) (cons(cons 'y y)(cons(cons 'x x)'())))) $\rightarrow$#f)

By M1 this is equivalent to:

$\forall$x(($\forall$y((mgs '(< x y) (cons(cons 'y y)(cons(cons 'x x)'()))) $\rightarrow$(mgs '#f (cons(cons 'y y)(cons(cons 'x x)'()))))) $\rightarrow$#f)

By M2 this is equivalent to: $\forall$x$\exists$y((mgs '(< x y) (cons(cons 'y y)(cons(cons 'x x)'()))) By M4 this is equivalent to: $\forall$x$\exists$y(<(mgn 'x(cons(cons 'y y)(cons(cons 'x x)'()))) (mgn 'y (cons(cons 'y y)(cons(cons 'x x)'()))))

By M6 twice this is equivalent to: $\forall$x$\exists$y(< x y)

The meaning of a set of sentences is defined in terms of the meanings of the sentences in the set as:

(ms 'κ) =df $\forall$s((sε'κ)$\rightarrow$(mg s))

MS1: (ms{'α: Γ}) $\leftrightarrow$ $\forall\xi$($\Gamma\rightarrow\alpha$) where ξ is the sequence of all the free variables in 'α and where Γ is any sentence of the intensional semantics.

proof: (ms{'α:Γ}) Unfolding ms and the set pattern abstraction symbol gives: $\forall$s((sε{s: $\exists\xi$((s='α)$\wedge\Gamma$)})$\rightarrow$(mg s))

where ξ is a sequence of the free variables in 'a. This is equivalent to: $\forall$s(($\exists\xi$((s='α)$\wedge\Gamma$)))$\rightarrow$(mg s))

which is logically equivalent to: $\forall$s$\forall\xi$ (((s='κ)$\wedge\Gamma$)$\rightarrow$(mg s)) which is equivalent to: $\forall\xi$($\Gamma\rightarrow$(mg 'α))

Unfolding mg using M0-M7 then gives: $\forall\xi$($\Gamma\rightarrow\alpha$) QED

The meaning of the union of two sets of FOL sentences is the conjunction of their meanings (i.e. MS1) and the meaning of a set is the meaning of all the sentences in the set (i.e. MS2):

MS2: (ms{'Γ_i}) $\leftrightarrow$ $\forall$i$\forall\xi_i\Gamma_i$

proof: (ms{'Γ_i}) Unfolding the set notation gives: (ms{'Γ_i: #t})

By MS1 this is equivalent to: $\forall_i\forall\xi_i$(#t$\rightarrow\Gamma_i$) which is equivalent to: $\forall$i$\forall\xi_i\Gamma_i$ QED.

MS3: (ms('κ∪'Γ)) ↔ ((ms 'κ)∧(ms 'Γ))

proof: Unfolding ms and union in: (ms('κ∪'Γ)) gives: ∀s((sε{s: (sε'κ)∨(sε'Γ)})→(mg s)) or rather: ∀s(((sε'κ)∨(sε'Γ))→(mg s)) which is logically equivalent to: (∀α((sε'κ)→(mg s)))∧(∀s((sε'Γ)→(mg s))) Folding ms twice then gives:((ms 'κ)∧(ms 'Γ)) QED.

The meaning operation may be used to develop an Intensional Semantics for a FOL object language by axiomatizing the modal concept of necessity so that it satisfies the theorem:

C1: ('αε(fol 'κ)) ↔ ([] ((ms 'κ)→(mg 'α)))

for every sentence 'α and every set of sentences 'κ of that FOL object language. The necessity symbol is represented by a box: []. C1 states that a sentence of FOL is a FOL-theorem (i.e. fol) of a set of sentences of FOL if and only if the meaning of that set of sentences necessarily implies the meaning of that sentence. One modal logic which satisfies C1 for FOL[4] is the Z Modal Quantificational Logic described in [Brown 1987; Brown 1989] whose theorems are recursively enumerable. Z has the metatheorem: (<>Γ){π/λξα}→ (<>Γ) where Γ is a sentence of FOL and includes all the laws of S5 Modal Logic [Hughes & Cresswell 1968] whose modal axioms and inference rules are given in Figure 6. Therein, κ and Γ represent arbitrary sentences of the intentional semantics.

R0: from α infer ([] κ)	A2: ([](κ→ Γ)) → (([]κ)→ ([]Γ))
A1: ([]κ) → κ	A3: ([]κ) ∨ ([]¬[]κ)

Figure 6: The Laws of S5 Modal Logic

These S5 modal laws and the laws of FOL given in Figure 4 constitute an S5 Modal Quantificational Logic similar to [Carnap 1946; Carnap 1956], and a FOL version [Parks 1976] of [Bressan 1972] in which the Barcan formula: (∀γ([]κ))→([]∀γκ) and its converse hold. The R0 inference rule implies that anything derivable in the metatheory is necessary. Thus, in any logic with R0, contingent facts would never be asserted as additional axioms of the metatheory. For example, we would not assert ([](κ↔Γ)) as an axiom and then try to prove ([](κ→α)). Instead we would try to prove that ([](κ↔Γ))→([](κ→α)).

The defined Modal symbols used herein are listed in Figure 7 with their definitions and interpretations.

Symbol	Definition	Meaning		Symbol	Definition	Meaning
<>κ	¬ [] ¬κ	α is logically possible		[κ] Γ	[] (κ→Γ)	β entails α
κ≡ Γ	[] (κ↔Γ)	α is logically equivalent to β		<κ> Γ	<> (κ∧Γ)	α and β is logically possible

Figure 7: Defined Symbols of Modal Logic

For example, folding the definition of entailment, C1 may be rewritten more compactly as:

C1': ('αε(fol 'κ)) ↔ ([(ms 'κ)](mg 'α))

This compact notation for entailment is used hereafter.

From the laws of the Intensional Semantics we prove that the meaning of the set of FOL consequences of a set of sentences is the meaning of that set of sentences (C2), the FOL consequences of a set of sentences contain the FOL consequences of another set if and only if the meaning of the first set entails the meaning of the second set (C3), and the sets of FOL consequences of two sets of sentences are equal if and only if the meanings of the two sets are logically equivalent (C4):

C2: (ms(fol 'κ))≡(ms 'κ)

proof: The proof divides into two cases:

(1) [(ms 'κ)](ms(fol 'κ))) Unfolding the second ms gives: [(ms 'κ)]∀s((sε(fol 'κ))→(mg s))

By the soundness part of C1 this is equivalent to: [(ms 'κ)]∀s(([(ms 'κ)](mg s))→(mg s))

By the S5 laws this is equivalent to: ∀s(([(ms 'κ)](mg s))→ [(ms 'κ)](mg s)) which is a tautology.

(2) [(ms(fol 'κ))](ms 'κ) Unfolding ms twice gives: [∀s((sε(fol 'κ))→(mg s))]∀s((sε'κ)→(mg s))

which is: [∀s((sε(fol 'κ))→(mg s))]((sε'κ)→(mg s)) Backchaining on the hypothesis and then dropping it gives: (sε'κ)→(sε(fol 'κ)). Folding ⊇ gives an instance of FOL1. QED.

[4] An S5 modal logic which satisfies a metatheorem analogous to C1 for Propositional Logic is the system S5c given in [Hendry and Pokriefka 1985] which has axiom schemes stating that every conjunction of distinct propositional constants is logically possible. This extends the trivial possibility axiom that some proposition is neither #t nor #f used in [Lewis 1936; Bressan 1972].

C3: $(fol\ '\kappa)\supseteq(fol\ '\Gamma) \leftrightarrow ([(ms\ '\kappa)](ms\ '\Gamma))$

proof: Unfolding $\supseteq$ gives: $\forall s((s\varepsilon(fol\ '\Gamma))\rightarrow(s\varepsilon(fol\ '\kappa)))$

By C1 twice this is equivalent to: $\forall s(([(ms\ '\Gamma)](mg\ s))\rightarrow([(ms\ '\kappa)](mg\ s)))$

By the laws of S5 modal logic this is equivalent to: $([(ms\ '\kappa)]\forall s(([(ms\ '\Gamma)](mg\ s)))\rightarrow(mg\ s)))$

By C1 this is equivalent to: $[(ms\ '\kappa)]\forall s((s\varepsilon(fol\ '\Gamma))\rightarrow(mg\ s))$. Folding ms then gives: $[(ms\ '\kappa)](ms(fol\ '\Gamma))$

By C2 this is equivalent to: $[(ms\ '\kappa)](ms\ '\Gamma)$. QED.

C4: $((fol\ '\kappa)=(fol\ '\Gamma)) \leftrightarrow ((ms\ '\kappa)\equiv(ms\ '\Gamma))$

proof: This is equivalent to $(((fol\ '\kappa)\supseteq(fol\ '\Gamma))\wedge((fol\ '\Gamma)\supseteq(fol\ '\kappa))) \leftrightarrow ([(ms\ '\kappa)](ms\ '\Gamma))\wedge([(ms\ '\Gamma)](ms\ '\kappa))$ which follows by using C3 twice.

5. Reflective Logic Represented in Modal Logic

The fixed-point equation for Reflective Logic may be expressed as a necessary equivalence in an S5 Modal Quantificational Logic as follows: $\kappa\equiv(RL\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)$ where RL is defined as: $(RL\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)$ =df $\Gamma\wedge\forall i((([\kappa]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow\chi_i)$ where Γ, α_i, β_{ij}, and χ_i are propositions of FOL. When the context is obvious $\Gamma\ \alpha_i{:}\beta_{ij}/\chi_i$ is omitted and just $(RL\ \kappa)$ is written. Given below are some simple properties of RL. The first two theorems state that RL entails Γ and any conclusion χ_i of a default whose entailment condition holds in κ and whose possible conditions are possible with κ.

MR1: $[(RL\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)]\Gamma$

proof: By R0 it suffices to prove: $(RL\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)\rightarrow\Gamma$. Unfolding RL gives: $\Gamma\wedge\forall_i((([\kappa]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow\chi_i)\rightarrow\Gamma$ which is a tautology. QED.

MR2: $(([\kappa]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow([(RL\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)]\chi_i)$

proof: Unfolding RL gives: $(([\kappa]\alpha_i)\wedge(\wedge_{j=1,m_i}(<\kappa>\beta_{ij})))\rightarrow([\Gamma\wedge\forall_i((([\kappa]\alpha_i)\wedge(\wedge_{j=1,m_i}(<\kappa>\beta_{ij})))\rightarrow\chi_i)]\chi_i)$

Using the hypotheses on the ith instance gives:

$(([\kappa]\alpha_i)\wedge(\wedge_{j=1,m_i}(<\kappa>\beta_{ij})))\rightarrow([\Gamma\wedge\forall_i((([\kappa]\alpha_i)\wedge(\wedge_{j=1,m_i}(<\kappa>\beta_{ij})))\rightarrow\chi_i)\wedge\chi_i]\chi_i)$ which is a tautology. QED.

The concept (i.e. ss) of the combined meaning of all the sentences of the FOL object language whose meanings are entailed by a proposition is defined as follows:

$(ss\ \kappa)$ =df $\forall s(([\kappa](mg\ s))\rightarrow(mg\ s))$

SS1 shows that a proposition entails the combined meaning of the FOL object language sentences that it entails. SS2 shows that if a proposition is necessarily equivalent to the combined meaning of the FOL object language sentences that it entails, then there exists a set of FOL object language sentences whose meaning is necessarily equivalent to that proposition:

SS1: $[\kappa](ss\ \kappa)$

proof: By R0 it suffices to prove: $\kappa\rightarrow(ss\ \kappa)$. Unfolding ss gives: $\kappa\rightarrow\forall s(([\kappa](mg\ s))\rightarrow(mg\ s))$ which is equivalent to:$\forall s(([\kappa](mg\ s))\rightarrow(\kappa\rightarrow(mg\ s)))$ which is an instance of A1. QED.

SS2: $(\kappa\equiv(ss\ \kappa))\rightarrow\exists s(\kappa\equiv(ms\ s))$

proof: Letting s be {s: $([\kappa](mg\ s))$ gives $(\kappa\equiv(ss\ \kappa))\rightarrow(\kappa\equiv(ms\{s{:}\ ([\kappa](mg\ s))))$. Unfolding ms and lambda conversion gives: $(\kappa\equiv(ss\ \kappa))\leftrightarrow(\kappa\equiv\forall s(([\kappa](mg\ s))\rightarrow(mg\ s)))$. Folding ss gives a tautology. QED.

The theorems MR3 and MR4 are analogous to MR1 and MR2 except that RL is replaced by the combined meanings of the sentences entailed by RL.

MR3: $[ss(RL\ \kappa\ \forall i\Gamma_i\ \alpha_i{:}\beta_{ij}/\chi_i)]\forall i\Gamma_i$

proof: By R0 it suffices to prove: $(ss(RL\ \kappa\ \forall i\Gamma_i\ \alpha_i{:}\beta_{ij}/\chi_i))\rightarrow\forall i\Gamma_l$ which is equivalent to:

$(ss(RL\ \kappa\ \forall i\Gamma_i\ \alpha_i{:}\beta_{ij}/\chi_i))\rightarrow\Gamma_i$. Unfolding ss gives: $(\forall s(([(RL\ \kappa\ \forall i\Gamma_i\ \alpha_i{:}\beta_{ij}/\chi_i)](mg\ s))\rightarrow(mg\ s)))\rightarrow\Gamma_i$

which by the meaning laws M0-M8 is equivalent to: $(\forall s(([(\mathrm{RL}\ \kappa\ \forall i\Gamma_i\ \alpha_i{:}\beta_{ij}/\chi_i)](\mathrm{mg}\ s))\rightarrow(\mathrm{mg}\ s)))\rightarrow(\mathrm{mg}\ '\Gamma_i)$
Backchaining on $(\mathrm{mg}\ \ '\Gamma_i)$ with s in the hypothesis being $'\Gamma_i$ in the conclusion shows that it suffices to prove:
$([(\mathrm{RL}\ \kappa\ \forall i\Gamma_i\ \alpha_i{:}\beta_{ij}/\chi_i)](\mathrm{mg}\ '\Gamma_i))$ which by the meaning laws: M0-M7 is equivalent to: $([(\mathrm{RL}\ \kappa\ \forall i\Gamma_i\ \alpha_i{:}\beta_{ij}/\chi_i)]\Gamma_i)$
which by the laws of S5 Modal Logic is equivalent to: $([(\mathrm{RL}\ \kappa\ \forall i\Gamma_i\ \alpha_i{:}\beta_{ij}/\chi_i)]\forall i\Gamma_i)$
which is an instance of theorem MR1. QED.

MR4: $(([\kappa]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow ([\mathrm{ss}(\mathrm{RL}\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)]\chi_i)$
proof: Unfolding the last ss gives: $(([\kappa]\alpha_i)\wedge\wedge_{j=1,mi}(<\kappa>\beta_{ij}))\rightarrow ([\forall s(([(\mathrm{RL}\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)](\mathrm{mg}\ s))\rightarrow(\mathrm{mg}\ s))]\chi_i)$
Instantiating s in the hypothesis to $'\chi_i$ and then dropping the hypothesis gives:
$(([\kappa]\alpha_i)\wedge\wedge_{j=1,mi}(<\kappa>\beta_{ij}))\rightarrow ([([(\mathrm{RL}\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)](\mathrm{mg}\ '\chi_i))\rightarrow(\mathrm{mg}\ '\chi_i)]\chi_i)$
Using the meaning laws M0-M7 gives: $(([\kappa]\alpha_i)\wedge\wedge_{j=1,mi}(<\kappa>\beta_{ij}))\rightarrow ([([(\mathrm{RL}\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)]\chi_i)\rightarrow\chi_i]\chi_i)$
Backchaining on χ_i shows that it suffices to prove:$(([\kappa]\alpha_i)\wedge\wedge_{j=1,mi}(<\kappa>\beta_{ij}))\rightarrow([(\mathrm{RL}\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)]\chi_i)$
which is an instance of theorem MR2. QED.

Finally MR5 and MR6 show that talking about the meanings of sets of FOL sentences in the modal representation of Reflective Logic is equivalent to talking about propositions in general.
MR5: $(\mathrm{ss}(\mathrm{RL}\ \kappa(\forall i\Gamma_i)\alpha_i{:}\beta_{ij}/\chi_i))\equiv(\mathrm{RL}\ \kappa(\forall i\Gamma_i)\alpha_i{:}\beta_{ij}/\chi_i)$ proof: In view of SS1, it suffices to prove
:($[(\mathrm{ss}(\mathrm{RL}\ \kappa(\forall i\Gamma_i)\alpha_i{:}\beta_{ij}/\chi_i))](\mathrm{RL}\ \kappa(\forall i\Gamma_i)\alpha_i{:}\beta_{ij}/\chi_i))$. Unfolding the second occurrence of RL gives:$[(\mathrm{ss}(\mathrm{RL}\ \kappa(\forall i\Gamma_i)\alpha_i{:}\beta_{ij}/\chi_i))](\forall i\Gamma_i\wedge\forall i((([\kappa]\alpha_i)\wedge\wedge_{j=1,mi}<\kappa>\beta_{ij})\rightarrow\chi_i)$ which holds by theorems MR3 and MR4. QED.

MR6: $(\kappa\equiv(\mathrm{RL}\ \kappa(\forall i\Gamma_i)\alpha_i{:}\beta_{ij}/\chi_i))\rightarrow\exists s(\kappa\equiv(\mathrm{ms}\ s))$
proof: From the hypothesis and MR5 $\kappa\equiv(\mathrm{ss}(\mathrm{RL}\ \kappa\ \forall i\Gamma_i\ \alpha_i{:}\beta_{ij}/\chi_i))$ is derived. Using the hypothesis to replace $(\mathrm{RL}\ \kappa(\forall i\Gamma_i)\alpha_i{:}\beta_{ij}/\chi_i)$ by κ in this result gives: $\kappa\equiv(\mathrm{ss}(\mathrm{RL}\ \kappa(\forall i\Gamma_i)\alpha_i{:}\beta_{ij}/\chi_i))$, By SS2 this implies the conclusion. QED.

6. Conclusion: The Relationship between Reflective Logic and the Modal Logic

The relationship between the proof theoretic definition of Reflective Logic [Brown 1989] and the modal representation is developed and proven in two steps. First theorem RL1 shows that the meaning of the set rl is the proposition RL and then theorem RL2 shows that a set of FOL sentences which contains its FOL theorems is a fixed-point of the fixed-point equation of Reflective Logic with an initial set of axioms and defaults if and only if the meaning (or rather disquotation) of that set of sentences is logically equivalent to RL of the meanings of that initial set of sentences and those defaults.

RL1: $(\mathrm{ms}(\mathrm{rl}(\mathrm{fol}\ '\kappa)\{'\Gamma_i\}'\alpha_i{:}'\beta_{ij}/'\chi_i))\equiv(\mathrm{RL}(\mathrm{ms}\ '\kappa)(\forall i\Gamma_i)\alpha_i{:}\beta_{ij}/\chi_i)$
proof: $(\mathrm{ms}(\mathrm{rl}(\mathrm{fol}\ '\kappa)\{'\Gamma_i\}'\alpha_i{:}'\beta_{ij}/'\chi_i))$
Unfolding the definition of rl gives: $\mathrm{ms}(\mathrm{fol}(\{'\Gamma_i\}\cup\{'\chi_i{:}('\alpha_i\varepsilon(\mathrm{fol}\ '\kappa))\wedge\wedge_{j=1,mi}('(\neg\beta_{ij})\not\in(\mathrm{fol}\ '\kappa))\}))$
By C2 this is equivalent to: $\mathrm{ms}(\{'\Gamma_i\}\cup\{'\chi_i{:}('\alpha_i\varepsilon(\mathrm{fol}\ '\kappa))\wedge\wedge_{j=1,mi}('(\neg\beta_{ij})\not\in(\mathrm{fol}\ '\kappa))\})$
Using C1 twice gives: $\mathrm{ms}(\{'\Gamma_i\}\cup\{'\chi_i{:}\ (([(\mathrm{ms}\ '\kappa)](\mathrm{mg}\ '\alpha_i))\ \wedge\wedge_{j=1,mi}\neg([(\mathrm{ms}\ '\kappa)](\mathrm{mg}\ '(\neg\beta_{ij}))))\})$
Using MS3 gives: $(\mathrm{ms}\ \{'\Gamma_i\})\wedge(\mathrm{ms}\{'\chi_i{:}\ (([(\mathrm{ms}\ '\kappa)](\mathrm{mg}\ '\alpha_i))\ \wedge\wedge_{j=1,mi}\neg([(\mathrm{ms}\ '\kappa)](\mathrm{mg}\ '(\neg\beta_{ij}))))\})$
Using MS2 gives: $(\forall i\Gamma_i)\wedge(\mathrm{ms}\{'\chi_i{:}\ (([(\mathrm{ms}\ '\kappa)](\mathrm{mg}\ '\alpha_i))\ \wedge\wedge_{j=1,mi}\neg([(\mathrm{ms}\ '\kappa)](\mathrm{mg}\ '(\neg\beta_{ij}))))\})$
Using MS1 gives: $(\forall i\Gamma_i)\wedge\forall_i((([(\mathrm{ms}\ '\kappa)](\mathrm{mg}\ '\alpha_i))\wedge\wedge_{j=1,mi}\neg([(\mathrm{ms}\ '\kappa)](\mathrm{mg}\ '(\neg\beta_{ij}))))\rightarrow(\mathrm{mg}\ '\chi_i))$
Using M0-M7 gives: $(\forall i\Gamma_i)\wedge\forall_i((([(\mathrm{ms}\ '\kappa)]\alpha_i)\wedge\wedge_{j=1,mi}\neg([(\mathrm{ms}\ '\kappa)]\neg\beta_{ij}))\rightarrow \chi_i)$
Folding the definition of RL then gives: $(\mathrm{RL}(\mathrm{ms}\ '\kappa)(\forall i\Gamma_i)\alpha_i{:}\beta_{ij}/\chi_i)$ QED.

RL2: $((\text{fol }'\kappa)=(\text{rl}(\text{fol }'\kappa)\{'\Gamma_i\}\ '\alpha_i:'\beta_{ij}/'\chi_i))\leftrightarrow((\text{ms }'\kappa)\equiv(\text{RL}(\text{ms }'\kappa)(\forall i\Gamma_i)\alpha_i:\beta_{ij}/\chi_i))$

proof: $(\text{fol }'\kappa)=(\text{rl}(\text{fol }'\kappa)\{'\Gamma_i\}\ '\alpha_i:'\beta_{ij}/'\chi_i)$ By FOL3 this is equivalent to: $(\text{fol }'\kappa)=(\text{fol}(\text{rl}(\text{fol }'\kappa)\{'\Gamma_i\}\ '\alpha_i:'\beta_{ij}/'\chi_i))$

By C4 this is equivalent to: $((\text{ms }'\kappa)\equiv(\text{ms}(\text{rl}(\text{fol }'\kappa)\{'\Gamma_i\}\ '\alpha_i:'\beta_{ij}/'\chi_i)))$

By RL1 this is equivalent to: $(\text{ms }'\kappa)\equiv(\text{RL}(\text{ms }'\kappa)(\forall i\Gamma_i)\ \alpha_i:\beta_{ij}/\chi_i))$ QED.

Theorem RL2 shows that the set of theorems: $(\text{fol }'\kappa)$ of a set $'\kappa$ is a fixed-point of a fixed-point equation of Reflective Logic if and only if the meaning $(\text{ms }'\kappa)$ of $'\kappa$ is a solution to the necessary equivalence. Furthermore, by FOL4 there are no other fixed-points (such as a set not containing all its theorems) and by MR6 there are no other solutions (such as a proposition not representable as a sentence in the FOL object language). Therefore, the Modal representation of Reflective Logic (i.e. RL), faithfully represents the set theoretic description of Reflective Logic (i.e. rl).Finally, we note that $\forall i\Gamma_i$ and $(\text{ms }'\kappa)$ may be generalized to be arbitrary propositions Γ and κ giving the more general modal representation: $\kappa\equiv(\text{RL }\kappa\ \Gamma\ \alpha_i:\beta_{ij}/\chi_i)$.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant Numbers: MIP-9526532, EIA-9818341, and EIA-9972843.

Bibliography

[Boyer&Moore 1981] R. S. Boyer and J. Strother Moore, "Metafunctions: proving them correct and using them efficiently as new proof procedures," *The Correctness Problem in Computer Science,* R. S. Boyer and J. Strother Moore, eds., Academic Press, New York, 1981.

[Bressan 1972] Bressan, Aldo 1972. *A General Interpreted Modal Calculus,* Yale University Press.

[Brown 1978] Brown,F.M., "A Semantic Theory for Logic Programming", *Colloquia Mathematica Societatis Janos Bolyai 26*, *Mathematical Logic in Computer Science*, Salgotarjan, Hungry, 1978.

[Brown 1987] Brown, Frank M. 1987. "The Modal Logic Z", In *The Frame Problem in AI"; Proc. of the 1987 AAAI Workshop*, Morgan Kaufmann, Los Altos, CA .

[Brown 1989] Brown, Frank M. 1989. "The Modal Quantificational Logic Z Applied to the Frame Problem", advanced paper *First International Workshop on Human & Machine Cognition*, May 1989 Pensacola, Florida. Abbreviated version published in *International Journal of Expert Systems Research and Applications, Special Issue: The Frame Problem. Part A.* eds. Keneth Ford and Pattrick Hayes, vol. 3 number 3, pp169-206 JAI Press 1990. Reprinted in *Reasoning Agents in a Dynamic World: The Frame problem*, editors: Kenneth M. Ford, Patrick J. Hayes, JAI Press 1991.

[Carnap 1946] Carnap, Rudolf 1946. "Modalities and Quantification" *Journal of Symbolic Logic*, vol. 11, number 2.

[Carnap 1956] Carnap, Rudolf 1956. *Meaning and Necessity: A Study in the Semantics of Modal Logic*, The University of Chicago Press.

[Hendry & Pokriefka 1985] Hendry, Herbert E. and Pokriefka, M. L. 1985. "Carnapian Extensions of S5", *Journal of Phil. Logic 14*.

[Hughes & Cresswell 1968] Hughes, G. E. & Creswell, M. J., 1968. *An Introduction to Modal Logic*, Methuen & Co. Ltd., London.

[Lewis 1936] Lewis, C. I. 1936. Strict Implication, *Journal of Symbolic Logic*, vol I.

[Mendelson 1964] Mendelson, E. 1964. *Introduction to Mathematical Logic*, Van Norstrand, Reinhold Co., New York.

[Moore 1985] Moore, R. C. 1985. "Semantical Considerations on Nonmonotonic Logic" *Artificial Intelligence*, 25.

[Parks 1976] Parks, Z. 1976. "An Investigation into Quantified Modal Logic", Studia Logica 35, p109-125.

[Quine 1969] Quine, W.V.O., Set Theory and Its Logic, revised edition, Oxford University Press, London, 1969.

[Reiter 1980] Reiter, R. 1980. "A Logic for Default Reasoning" *Artificial Intelligence*, 13.

Author information

Frank M. Brown- Artificial Intelligence Laboratory, University of Kansas, Lawrence, Kansas, 66045, e-mail: brown@ku.edu.

REPRESENTING DEFAULT LOGIC IN MODAL LOGIC

Frank M. Brown

Abstract: The nonmonotonic logic called Default Logic is shown to be representable in a monotonic Modal Quantificational Logic whose modal laws are stronger than S5. Specifically, it is proven that a set of sentences of First Order Logic is a fixed-point of the fixed-point equation of Default Logic with an initial set of axioms and defaults if and only if the meaning or rather disquotation of that set of sentences is logically equivalent to a particular modal functor of the meanings of that initial set of sentences and of the sentences in those defaults. This result is important because the modal representation allows the use of powerful automatic deduction systems for Modal Logic and because unlike the original Default Logic, it is easily generalized to the case where quantified variables may be shared across the scope of the components of the defaults thus allowing such defaults to produce quantified consequences. Furthermore, this generalization properly treats such quantifiers since both the Barcan Formula and its converse hold.

Keywords: Default Logic, Modal Logic, Nonmonotonic Logic.

1. Introduction

One of the most well known nonmonotonic logics [Antoniou 1997] which inherently deals with entailment conditions in addition to possibility conditions in its defaults is the so-called Default Logic [Reiter 1980]. The basic idea of Default Logic is that there is a set of axioms Γ and some non-logical default "inference rules" of the form:

$$\frac{\alpha : \beta_1,\ldots,\beta_m}{\chi}$$

which suggest that χ may be inferred from α whenever each $\beta_1,\ldots,\beta_m$ is consistent with everything that is inferable. Such "inference rules" are not recursive and are circular in that the determination as to whether χ is derivable depends on whether β_j is consistent which in turn depends on what was derivable from this and other defaults. Thus, tentatively applying such inference rules by checking the consistency of $\beta_1,\ldots,\beta_m$ with only the current set of inferences produces a χ result which may later have to be retracted. For this reason, valid inferences in a nonmonotonic logic such as Default Logic are essentially carried out not in the original nonmonotonic logic, but rather in some (monotonic) metatheory in which that nonmonotonic logic is defined. [Reiter 1980] explicated this intuition by defining Default Logic in terms of the set theoretic proof theory metalanguage of First Order Logic (i.e. FOL) with the following fixed-point expression: $'\kappa=(\mathrm{dl}\ '\kappa\ \{'\Gamma_i\}\ '\alpha_i:'\beta_{ij}/'\chi_i)$ where dl is: $(\mathrm{dl}\ '\kappa\ \{'\Gamma_i\}\ '\alpha_i:'\beta_{ij}/'\chi_i) =\mathrm{df} \cap\{p: (p\supseteq(\mathrm{fol}\ p))\wedge(p\supseteq\{'\Gamma_i\})\wedge\forall_i((('\alpha_i\varepsilon p)\wedge\wedge_{j=1,m_i}('(\neg\beta_{ij})\not\in '\kappa))\rightarrow('\chi_i\varepsilon p))\}$ where $'\alpha_i$, $'\beta_{ij}$, and $'\chi_i$ are the closed sentences of FOL occurring in the ith default "inference rule" and $\{'\Gamma_i\}$ is a set of closed sentences of FOL. A closed sentence is a sentence without any free variables. fol is a function which produces the set of theorems derivable in FOL from the set of sentences to which it is applied. The quotations appended to the front of these Greek letters indicate references in the metalanguage to the sentences of the FOL object language. Interpreted doxastically this fixed-point equation states:

The set of closed sentences which are believed is equal to
the intersection of all sets of closed sentences which are potentialially believed such that:
the closed sentences derived by the laws of FOL from the potential beliefs are themselves potentially believed,
the closed sentences in $\{'\Gamma i\}$ are potentially believed,
and for each i, if the closed sentence $'\alpha_i$ is potentially believed
and for each j, the closed sentence $'\beta_{ij}$ is believable then the closed sentence $'\chi_i$ is potentially believed.

The purpose of this paper is to show that all this metatheoretic machinery including the formalized syntax of FOL, the proof theory of FOL, the axioms of a strong set theory, and the set theoretic fixed-point equation is not needed and that the essence of Default Logic is representable as a necessary equivalence in a simple (monotonic) Modal Quantificational Logic. Interpreted as a doxastic logic this necessary equivalence states:

That which is believed is logically equivalent to some potential belief such that:

Γ is potentially believed

and for each i, if α_i is potentially believed and for each j, β_{ij} is believable then χ_i is potentially believed.

thereby eliminating all mention of any metatheoretic machinery.

The remainder of this paper proves that this modal representation is equivalent to Default Logic. Section 2 describes a formalized syntax for a FOL object language. Section 3 describes the part of the proof theory of FOL needed herein (i.e. theorems FOL1-FOL9). Section 4 describes the Intensional Semantics of FOL including the meaning operator (i.e. the laws M0-M7) and the relationship of meaning and modality to the proof theory of FOL (i.e. the laws R0, A1, A2 and A3 and the theorems C1, C2, C3, and C4). The modal version of Default Logic, called DL, is defined in section 5 and explicated with theorems MD1-MD7 and SS1-SS2. In section 6, this modal version is shown by theorems DL1 and DL2 to be equivalent to the set theoretic fixed-point equation for Default Logic. Figure 1 outlines the relationship of all these theorems to the final theorems DL2, FOL9, and MD7.

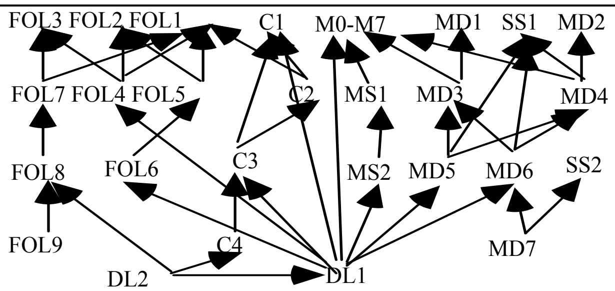


Figure 1: Dependencies among the Theorems

2. Formal Syntax of First Order Logic

We use a First Order Logic (i.e. FOL) defined as the six tuple: ($\rightarrow$, #f, $\forall$, *vars*, *predicates*, *functions*) where $\rightarrow$, #f, and $\forall$ are logical symbols, *vars* is a set of variable symbols, *predicates* is a set of predicate symbols each of which has an implicit arity specifying the number of associated terms, and *functions* is a set of function symbols each of which has an implicit arity specifying the number of associated terms. The sets of logical symbols, variables, predicate symbols, and function symbols are pairwise disjoint. Lower case Roman letters possibly indexed with digits are used as variables. Greek letters possibly indexed with digits are used as syntactic metavariables. γ, γ_1,...γ_n, range over the variables, ξ, ξ_1...ξ_n range over sequences of variables of an appropriate arity, π,π_1...π_n range over the predicate symbols, ϕ, ϕ_1...ϕ_n range over function symbols, δ, δ_1...δ_n, σ range over terms, and α,α_1...α_n, β, β_1...β_n,χ, χ_1...χ_n, Γ_1,...Γ_n,φ range over sentences. The terms are of the forms γ and (ϕ δ1...δn), and the sentences are of the forms ($\alpha\rightarrow\beta$), #f, ($\forall\gamma$ α), and (π δ_1...δ_n). A nullary predicate π or function ϕ is written as a sentence or a term without parentheses. $\varphi\{\pi/\lambda\xi\alpha\}$ represents the replacement of all occurrences of π in φ by $\lambda\xi\alpha$ followed by lambda conversion. The primitive symbols are shown in Figure 2 with their intuitive interpretations.

Symbol	Meaning
$\alpha\rightarrow\beta$	if α then β.
#f	falsity
$\forall\gamma\ \alpha$	for all γ, α.

Figure 2: Primitive Symbols of First Order Logic

The defined symbols are listed in Figure 3 with their definitions and intuitive interpretations.

Symbol	Definition	Meaning		Symbol	Definition	Meaning
$\neg\alpha$	$\alpha\rightarrow$ #f	not α		$\alpha\wedge\beta$	$\neg(\alpha\rightarrow\neg\beta)$	α and β
#t	$\neg$ #f	truth		$\alpha\leftrightarrow\beta$	$(\alpha\rightarrow\beta)\wedge(\beta\rightarrow\alpha)$	α if and only if β
$\alpha\vee\beta$	$(\neg\alpha)\rightarrow\beta$	α or β		$\exists\gamma\ \alpha$	$\neg\forall\gamma\ \neg\alpha$	for some γ , α

Figure 3: Defined Symbols of First Order Logic

The FOL object language expressions are referred in the metalanguage (which also includes a FOL syntax) by inserting a quote sign in front of the object language entity thereby making a structural descriptive name of that entity. A set of sentences is represented as: {'Γ_i} which is defined as: {'Γ_i: #t} which in turn is defined as: {s: ∃i(s='Γ_i)} where i ranges over some range of numbers (which may be finite or non-infinite). With a slight abuse of notation we also write 'κ, 'Γ to refer to such sets.

3. Proof Theory of First Order Logic

First Order Logic (i.e. FOL) is axiomatized with a recursively enumerable set of theorems as the set of axioms is itself recursively enumerable and its inference rules are recursive. The axioms and inference rules of FOL [Mendelson 1964] are those given in Figure 4. They form a standard set of axioms and inference rules for FOL.

MA1: α → (β→ α)	MR1: from α and (α→ β) infer β
MA2: (α→ (β→ ρ)) → ((α→ β)→ (α→ ρ))	MR2: from α infer (∀γ α)
MA3: ((¬ α)→ (¬ β))→ (((¬ α)→ β)→α)	
MA4: (∀γ α)→ β where β is the result of substituting an expression (which is free for the free positions of γ in α) for all the free occurrences of γ in α.	
MA5: (∀γ(α → β)) → (α→(∀γ β)) where γ does not occur in α.	

Figure 4: Inferences Rules and Axioms of FOL

In order to talk about sets of sentences we include in the metatheory set theory symbolism as developed along the lines of [Quine 1969]. This set theory includes the symbols ε, ∉, ⊒, =, ∪ as is defined therein.

The derivation operation (i.e. fol) of any First Order Logic obeys the Inclusion (i.e. FOL1), Idempotence (i.e. FOL2), and Monotonic (i.e. FOL3) properties:

FOL1: (fol 'κ)⊒'κ Inclusion

FOL2: (fol 'κ)⊒(fol(fol 'κ)) Idempotence

FOL3: ('κ⊒'Γ) → ((fol 'κ)⊒ (fol 'Γ)) Monotonicity

From these three properties we prove the following theorems of the proof theory of First Order Logic:

FOL4 ((fol 'κ)⊒(fol 'Γ))↔((fol 'κ)⊒'Γ) proof: The proof divides into two parts: (1) ((fol 'κ)⊒(fol 'Γ)) → ((fol 'κ)⊒'Γ). By FOL1 the hypothesis implies the conclusion. (2) ((fol 'κ)⊒'Γ)→((fol 'κ)⊒(fol 'Γ)) By FOL3 the hypothesis implies (fol(fol 'κ))⊒(fol 'Γ) which by FOL2 implies the conclusion. QED.

FOL5: ∀p((p=(fol p))→α)↔∀p(α{p/(fol p)}) and ∃p((p=(fol p))∧α)↔∃p(α{p/(fol p)})

proof: The universal quantifier version follows from the existential quantifier version by running negation through both sides of the bi-implication. The existential version is proven as follows. There are two cases:

(1) ((p=(fol p))∧α)→ ∃p(α{p/(fol p)}). The existentially quantified p is replaced by p giving:

((p=(fol p))∧α)→(α{p/(fol p)}) The the hypothesis is used to replace p in α by (fol p) giving the conclusion.

(2) (α{p/(fol p)})→ ∃p((p=(fol p))∧α) Letting p in the conclusion be (fol p) gives:

(α{p/(fol p)})→ (((fol p)=(fol(fol p)))∧(α{p/(fol p)})) which holds by FOL1 and FOL2.

FOL6: (∩{p: (p⊒(fol p))∧φ})={s:∀p((φ{p/(fol p)})→(sε(fol p)))} proof: ∩{p: (p⊒(fol p))∧φ} By FOL1 this is equivalent to: ∩{p: (p=(fol p))∧φ}. Unfolding the definition of intersection gives:{s:∀p((pε{p: (p=(fol p))∧φ})→(sεp))} which is equivalent to: {s:∀p(((p=(fol p))∧φ)→(sεp))}. By FOL5 this is equivalent to: {s:∀p((φ{p/(fol p)})→(sε(fol p)))} QED.

FOL7: If α is a sentence of proof theory then: (∩{p: (p⊒(fol p))∧α})=(fol(∩{p: (p⊒(fol p))∧α}))

proof: From FOL1 it suffices to prove: (sε(fol(∩{p: (p⊒(fol p))∧α})))→(sε(∩{p: (p⊒(fol p))∧α})). Unfolding the intersections and simplifying gives: (sε(fol{s: ∀p(((p⊒(fol p))∧α)→(sεp))}))→∀p(((p⊒(fol p))∧α)→(sεp))which is equivalent to: ((sε(fol{s:(sεp)∧∀p(((p⊒(folp))∧α)→(sεp))}))∧(p⊒(fol p))∧α)→(sεp). Folding intersection then gives: ((sε(fol({s:(sεp)}∩{s:∀p(((p⊒(folp))∧α)→(sεp))})))∧(p⊒(fol p))∧α)→(sεp). Using the second hypothesis to replace p by (fol p) and then dropping the second and third hypotheses gives: (sε(fol(p∩{s:∀p(((p⊒(folp))∧α) →(sεp))})))→(sε(fol p)). Folding ⊒ gives: (fol p)⊒(fol(p∩{s:∀p(((p⊒(fol p))∧α) →(sεp))})). Generalizing, it suffices to prove for all α: (fol p)⊒(fol(p∩α)). Since p⊒(p∩α) this follows by FOL3. QED.

FOL8: (dl 'κ 'Γ 'α_i:'β_{ij}/'χ_i)=(fol(dl 'κ 'Γ 'α_i:'β_{ij}/'χ_i)) proof: Unfolding dl gives:∩{p: (p⊒(fol p))∧(p⊒'Γ)∧

$\forall_i((('\alpha_i\varepsilon p)\wedge \wedge_{j=1,mi}('(\neg\beta_{ij})\not\in'\kappa))\rightarrow('\chi_i\varepsilon p))\}$. By FOL7 this is equivalent to: $fol(\cap\{p: (p\supseteq(fol\ p))\wedge(p\supseteq'\Gamma)\wedge\forall_i((('\alpha_i\varepsilon p)\wedge\wedge_{j=1,mi}('(\neg\beta_{ij})\not\in'\kappa))\rightarrow('\chi_i\varepsilon p))\})$ Folding dl then proves the theorem: $fol(dl\ '\kappa\ '\Gamma\ '\alpha_i{:}'\beta_{ij}/'\chi_i)$ QED.

FOL9: $('k=(dl\ '\kappa\ '\Gamma\ '\alpha_i{:}'\beta_{ij}/'\chi_i))\rightarrow('\kappa=(fol\ '\kappa))$ proof: From the hypothesis and FOL8 $'k=(fol(dl\ '\kappa\ '\Gamma\ '\alpha_i{:}'\beta_{ij}/'\chi_i))$ is derived. Using the hypothesis to replace $(dl\ '\kappa\ '\Gamma\ '\alpha_i{:}'\beta_{ij}/'\chi_i)$ by $'\kappa$ in this result gives: $('\kappa=(fol\ '\kappa))$ QED.

4. Intensional Semantics of FOL

The meaning (i.e. mg) [Brown 1978, Boyer&Moore 1981] or rather disquotation of a sentence of First Order Logic (i.e. FOL) is defined to satisfy the laws given in Figure 5 below . mg is defined in terms of mgs which maps a FOL object language sentence and an association list into a meaning. Likewise, mgn maps a FOL object language term and an association list into a meaning. An association list is a list of pairs consisting of an object language variable and the meaning to which it is bound.

M0: $(mg\ '\alpha)$ =df $(mgs\ '(\forall\gamma 1...\gamma n\ \alpha)'())$ where $'\gamma 1...'\gamma n$ are all the free variables in $'\alpha$

M1: $(mgs\ '(\alpha\rightarrow\beta)a)\leftrightarrow((mgs\ '\alpha\ a)\rightarrow(mgs\ '\beta\ a))$

M2: $(mgs\ '\#f\ a)\leftrightarrow\#f$

M3: $(mgs\ '(\forall\ \gamma\ \alpha)a)\leftrightarrow\forall x(mgs\ '\alpha(cons(cons\ '\gamma\ x)a))$

M4: $(mgs\ '(\pi\ \delta_1...\delta_n)a)\leftrightarrow(\pi(mgn\ '\delta_1\ a)...(mgn\ '\delta_n\ a))$ for each predicate symbol $'\pi$

M5: $(mgn\ '(\phi\ \ \delta_1...\delta_n)a)=(\phi(mgn\ '\delta_1\ a)...(mgn\ '\delta_n\ a))$ for each function symbol $'\phi$

M6: $(mgn\ '\gamma\ a)=(cdr(assoc\ '\gamma\ a))$

M7: (assoc v L) = (if(eq? v(car(car L))) (car L) (assoc v(cdr L)))
where: cons, car, cdr, eq?, if are axiomatized as they are axiomatized in Scheme.

Figure 5: The Meaning of FOL Sentences

The meaning of a set of sentences is defined in terms of the meanings of the sentences in the set as:

$(ms\ '\kappa)$ =df $\forall s((s\varepsilon'\kappa)\rightarrow(mg\ s))$

MS1: $(ms\{'\alpha{:}\ \Gamma\})\leftrightarrow\forall\xi(\Gamma\rightarrow\alpha)$ where ξ is the sequence of all the free variables in $'\alpha$ and where Γ is any sentence of the intensional semantics. proof: $(ms\{'\alpha{:}\Gamma\})$ Unfolding ms and the set pattern abstraction symbol gives: $\forall s((s\varepsilon\{s{:}\ \exists\xi((s='\alpha)\wedge\Gamma)\})\rightarrow(mg\ s))$ where ξ is a sequence of the free variables in 'a. This is equivalent to: $\forall s((\exists\xi((s='\alpha)\wedge\Gamma)))\rightarrow(mg\ s))$ which is logically equivalent to: $\forall s\forall\xi\ (((s='\kappa)\wedge\Gamma)\rightarrow(mg\ s))$ which is equivalent to: $\forall\xi(\Gamma\rightarrow(mg\ '\alpha))$ Unfolding mg using M0-M7 then gives: $\forall\xi(\Gamma\rightarrow\alpha)$ QED

The meaning of the union of two sets of FOL sentences is the conjunction of their meanings (i.e. MS3) and the meaning of a set is the meaning of all the sentences in the set (i.e. MS2):

MS2: $(ms\{'\Gamma_i\})\leftrightarrow\forall i\forall\xi_i\Gamma_i$ proof: $(ms\{'\Gamma_i\})$ Unfolding the set notation gives: $(ms\{'\Gamma_i{:}\ \#t\})$

By MS1 this is equivalent to: $\forall_i\forall\xi_i(\#t\rightarrow\Gamma_i)$ which is equivalent to: $\forall i\forall\xi_i\Gamma_i$ QED.

MS3: $(ms('\kappa\cup'\Gamma))\leftrightarrow((ms\ '\kappa)\wedge(ms\ '\Gamma))$ proof: Unfolding ms and union in: $(ms('\kappa\cup'\Gamma))$ gives: $\forall s((s\varepsilon\{s{:}\ (s\varepsilon'\kappa)\vee(s\varepsilon'\Gamma)\})\rightarrow(mg\ \ s))$ or rather: $\forall s(((s\varepsilon'\kappa)\vee(s\varepsilon'\Gamma))\rightarrow(mg\ \ s))$ which is logically equivalent to: $(\forall\alpha((s\varepsilon'\kappa)\rightarrow(mg\ s)))\wedge(\forall s((s\varepsilon'\Gamma)\rightarrow(mg\ s)))$. Folding ms twice then gives:$((ms\ '\kappa)\wedge(ms\ '\Gamma))$ QED.

The meaning operation may be used to develop an Intensional Semantics for a FOL object language by axiomatizing the modal concept of necessity so that it satisfies the theorem:

C1: $('\alpha\varepsilon(fol\ '\kappa))\ \leftrightarrow\ ([]\ ((ms\ '\kappa)\rightarrow(mg\ '\alpha)))$

for every sentence $'\alpha$ and every set of sentences $'\kappa$ of that FOL object language. The necessity symbol is represented by a box: []. C1 states that a sentence of FOL is a FOL-theorem (i.e. fol) of a set of sentences of FOL if and only if the meaning of that set of sentences necessarily implies the meaning of that sentence. One modal logic which satisfies C1 is the Z Modal Quantificational Logic described in [Brown 1987; Brown 1989] whose theorems are recursively enumerable and which extends the weaker possibility axioms used in [Lewis 1936; Bressan 1972; Hendry & Pokriefka 1985]. Z includes all the laws of S5 modal Logic [Hughes & Cresswell 1968] whose laws are given in Figure 6. κ and Γ represent arbitrary sentences of the intentional semantics.

R0: from α infer $([]\ \kappa)$	A2: $([](\kappa\rightarrow\Gamma))\rightarrow(([]\kappa)\rightarrow([]\Gamma))$
A1: $([]\kappa)\rightarrow\kappa$	A3: $([]\kappa)\vee([]\neg[]\kappa)$

Figure 6: The Laws of S5 Modal Logic

These S5 modal laws and the laws of FOL given in Figure 4 constitute an S5 Modal Quantificational Logic similar to [Carnap 1946; Carnap 1956], and a FOL version [Parks 1976] of [Bressan 1972] in which the Barcan formula: $(\forall\gamma([]\kappa))\rightarrow([]\forall\gamma\kappa)$ and its converse hold. The R0 inference rule implies that anything derivable in the metatheory is necessary. Thus, in any logic with R0, contingent facts would never be asserted as additional axioms of the metatheory. The defined Modal symbols used herein are listed in Figure 7.

Symbol	Definition	Meaning		Symbol	Definition	Meaning
$<>\kappa$	$\neg\ []\ \neg\kappa$	α is logically possible		$[\kappa]\ \Gamma$	$[]\ (\kappa\rightarrow\Gamma)$	β entails α
$\kappa\equiv\Gamma$	$[]\ (\kappa\leftrightarrow\Gamma)$	α is logically equivalent to β		$<\kappa>\ \Gamma$	$<>\ (\kappa\wedge\Gamma)$	α and β is logically possible

Figure 7: Defined Symbols of Modal Logic

From the laws of the Intensional Semantics we prove that the meaning of the set of FOL consequences of a set of sentences is the meaning of that set of sentences (C2), the FOL consequences of a set of sentences contain the FOL consequences of another set if and only if the meaning of the first set entails the meaning of the second set (C3), and the sets of FOL consequences of two sets of sentences are equal if and only if the meanings of the two sets are logically equivalent (C4):

C2: $(\mathrm{ms}(\mathrm{fol}\ '\kappa))\equiv(\mathrm{ms}\ '\kappa)$ proof: The proof divides into two cases: (1) $[(\mathrm{ms}\ '\kappa)](\mathrm{ms}(\mathrm{fol}\ '\kappa))$. Unfolding the second ms gives: $[(\mathrm{ms}\ '\kappa)]\forall s((s\varepsilon(\mathrm{fol}\ '\kappa))\rightarrow(\mathrm{mg}\ s))$. By the soundness part of C1 this is equivalent to: $[(\mathrm{ms}\ '\kappa)]\forall s(([(\mathrm{ms}\ '\kappa)](\mathrm{mg}\ s))\rightarrow(\mathrm{mg}\ s))$. By the S5 laws this is e: $\forall s(([(\mathrm{ms}\ '\kappa)](\mathrm{mg}\ s))\rightarrow [(\mathrm{ms}\ '\kappa)](\mathrm{mg}\ s))$ which is a tautology.

(2) $[(\mathrm{ms}(\mathrm{fol}\ '\kappa))](\mathrm{ms}\ '\kappa)$ Unfolding ms twice gives: $[\forall s((s\varepsilon(\mathrm{fol}\ '\kappa))\rightarrow(\mathrm{mg}\ s))]\forall s((s\varepsilon'\kappa)\rightarrow(\mathrm{mg}\ s))$

which is: $[\forall s((s\varepsilon(\mathrm{fol}\ '\kappa))\rightarrow(\mathrm{mg}\ s))]((s\varepsilon'\kappa)\rightarrow(\mathrm{mg}\ s))$ Backchaining on the hypothesis and then dropping it gives: $(s\varepsilon'\kappa)\rightarrow(s\varepsilon(\mathrm{fol}\ '\kappa))$. Folding $\supseteq$ gives an instance of FOL1. QED.

C3: $(\mathrm{fol}\ '\kappa)\supseteq(\mathrm{fol}\ '\Gamma)\leftrightarrow([(\mathrm{ms}\ '\kappa)](\mathrm{ms}\ '\Gamma))$

proof: Unfolding $\supseteq$ gives: $\forall s((s\varepsilon(\mathrm{fol}\ '\Gamma))\rightarrow(s\varepsilon(\mathrm{fol}\ '\kappa)))$.By C1 twice this is:$\forall s(([(\mathrm{ms}\ '\Gamma)](\mathrm{mg}\ s))\rightarrow([(\mathrm{ms}\ '\kappa)](\mathrm{mg}\ s)))$

By the laws of S5 modal logic this is equivalent to: $([(\mathrm{ms}\ '\kappa)]\forall s(([(\mathrm{ms}\ '\Gamma)](\mathrm{mg}\ s))\rightarrow(\mathrm{mg}\ s)))$. By C1 this is: $[(\mathrm{ms}\ '\kappa)]\forall s((s\varepsilon(\mathrm{fol}\ '\Gamma))\rightarrow(\mathrm{mg}\ s))$. Folding ms then gives: $[(\mathrm{ms}\ '\kappa)](\mathrm{ms}(\mathrm{fol}\ '\Gamma))$. By C2 this is : $[(\mathrm{ms}\ '\kappa)](\mathrm{ms}\ '\Gamma)$. QED.

C4: $((\mathrm{fol}\ '\kappa)=(\mathrm{fol}\ '\Gamma))\leftrightarrow((\mathrm{ms}\ '\kappa)\equiv(\mathrm{ms}\ '\Gamma))$ proof: This is equivalent to $(((\mathrm{fol}\ '\kappa)\supseteq(\mathrm{fol}\ '\Gamma))\wedge((\mathrm{fol}\ '\Gamma)\supseteq(\mathrm{fol}\ '\kappa)))\leftrightarrow([(\mathrm{ms}\ '\kappa)](\mathrm{ms}\ '\Gamma))\wedge([(\mathrm{ms}\ '\Gamma)](\mathrm{ms}\ '\kappa))$ which follows by using C3 twice.

5. Default Logic Represented in Modal Logic

The fixed-point equation for Default Logic may be expressed as a necessary equivalence in an S5 Modal Quantificational Logic supplemented with propositional quantifiers [Fine 1970; Bressan 1972] which obey the normal laws of Second Order Logic (i.e. laws analogous to MR2, MA4, and MA5 given in Figure 4 where γ is now a propositional variable), as follows: $\kappa\equiv(\mathrm{DL}\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)$

where DL is defined as: $(\mathrm{DL}\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)=\mathrm{df}\ \exists p(p\wedge([p]\Gamma)\wedge\forall_i((([p]\alpha_i)\wedge\wedge_{j=1,mi}(<\kappa>\beta_{ij}))\rightarrow[p]\chi_i))$

where the propositional variable p does not occur in Γ, α_i, β_{ij}, and χ_i. When the context is obvious $\Gamma\ \alpha_i{:}\beta_{ij}/\chi_i$ is omitted and just $(\mathrm{DL}\ \kappa)$ is written. The idiom $\exists p(p\wedge([]\varphi))$ may be intuitively read as a nominal as the (possibly infinite) disjunction of all propositions such that $[]\varphi$. When $[]\varphi$ holds for only a finite number of propositions: $\varphi_1,\ldots,\varphi_n$ then $\exists p(p\wedge([]\varphi))$is equivalent to: $\varphi_1\vee\ldots\vee\varphi_n$, but there is in no requirement that φ holds for only a finite or even only a denumerable number of propositions.

The first two theorems state that DL entails Γ and any conclusion χ_i of a default whose entailment condition holds in DL and whose possible conditions are possible with κ.

MD1: $[(\mathrm{DL}\ \kappa\ \ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)]\Gamma$

proof: Unfolding DL gives: $[\exists p(p\wedge([p]\Gamma)\wedge\forall_i((([p]\alpha_i)\wedge\wedge_{j=1,mi}(<\kappa>\beta_{ij}))\rightarrow([p]\chi_i)))]\Gamma$. Since p is not free in Γ, pulling $\exists p$ out of the hypothesis of the entailment gives:

$\forall p((([p]\Gamma)\wedge\forall_i((([p]\alpha_i)\wedge\wedge_{j=1,mi}<\kappa>\beta_{ij})\rightarrow([p]\chi_i)))\rightarrow([p]\Gamma))$ which is a tautology. QED.

MD2: $(([(DL\ \kappa\ \ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)]\alpha_i)\wedge\wedge_{j=1,mi}(<\kappa>\beta_{ij})) \rightarrow ([(DL\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)]\chi_i)$

proof: Unfolding both occurrences of DL gives:

$(([\exists p(p\wedge([p]\Gamma)\wedge\forall_i((([p]\alpha_i)\wedge\wedge_{j=1,mi}<\kappa>\beta_{ij})\rightarrow([p]\chi_i)))]\alpha_i)\wedge(\wedge_{j=1,m_i}(<\kappa>\beta_{ij})))$

$\rightarrow([\exists p(p\wedge([p]\Gamma)\wedge\forall_i((([p]\alpha_i)\wedge\wedge_{j=1,mi}(<\kappa>\beta_{ij}))\rightarrow([p]\chi_i)))]\chi_i)$

Since p is not free in α_i and χ_i, pulling $\exists p$ out of the hypotheses of the outer two entailments gives:

$((\forall p((([p]\Gamma)\wedge\forall_i((([p]\alpha_i)\wedge\wedge_{j=1,mi}(<\kappa>\beta_{ij}))\rightarrow([p]\chi_i)))\rightarrow([p]\alpha_i)))\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))$

$\rightarrow\forall p((([p]\Gamma)\wedge\wedge_i((([p]\alpha_i)\wedge\wedge_{j=1,mi}(<\kappa>\beta_{ij}))\rightarrow([p]\chi_i)))\rightarrow([p]\chi_i))$

Instantiating the p in the hypothesis to the p in the conclusion gives:

$(((([p]\Gamma)\wedge\forall_i((([p]\alpha_i)\wedge\wedge_{j=1,mi}<\kappa>\beta_{ij})\rightarrow([p]\chi_i)))\rightarrow([p]\alpha_i))\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij}))\wedge([p]\Gamma)\wedge\forall_i((([p]\alpha_i)\wedge\wedge_{j=1,mi}(<\kappa> \beta_{ij}))\rightarrow([p]\chi_i))) \rightarrow([p]\chi_i)$

which simplifies to just: $(([p]\alpha_i)\wedge\wedge_{j=1,mi}(<\kappa>\beta_{ij})\wedge([p]\Gamma)\wedge\forall_i((([p]\alpha_i)\wedge\wedge_{j=1,mi}(<\kappa>\beta_{ij}))\rightarrow([p]\chi_i))) \rightarrow([p]\chi_i)$

Forward chaining using the first and second hypotheses on the fourth proves the theorem. QED.

The concept (i.e. ss) of the combined meaning of all the sentences of the FOL object language whose meanings are entailed by a proposition is defined as follows: $(ss\ \kappa)$ =df $\forall s(([\kappa](mg\ s))\rightarrow(mg\ s))$. SS1 shows that a proposition entails the combined meaning of the FOL object language sentences that it entails. SS2 shows that if a proposition is necessarily equivalent to the combined meaning of the FOL object language sentences that it entails, then there exists a set of FOL object language sentences whose meaning is necessarily equivalent it:

SS1: $[\kappa](ss\ \kappa)$

proof: By R0 it suffices to prove: $\kappa\rightarrow(ss\ \kappa)$. Unfolding ss gives: $\kappa\rightarrow\forall s(([\kappa](mg\ s))\rightarrow(mg\ s))$

which is equivalent to:$\forall s(([\kappa](mg\ s))\rightarrow(\kappa\rightarrow(mg\ s)))$ which is an instance of A1. QED.

SS2: $(\kappa\equiv(ss\ \kappa))\rightarrow \exists s(\kappa\equiv(ms\ s))$

proof: Letting s be {s: $([\kappa](mg\ s))$ gives $(\kappa\equiv(ss\ \kappa))\rightarrow (\kappa\equiv(ms\{s{:}\ ([\kappa](mg\ s))))$. Unfolding ms and lambda conversion gives: $(\kappa\equiv(ss\ \kappa))\leftrightarrow (\kappa\equiv\forall s(([\kappa](mg\ s))\rightarrow(mg\ s)))$. Folding ss gives a tautology. QED.

The theorems MD3 and MD4 are analogous to MD1 and MD2 except that DL is replaced by the combined meanings of the sentences entailed by DL.

MD3: $[ss(DL\ \kappa\ \forall i\Gamma_i\ \alpha_i{:}\beta_{ij}/\chi_i)]\forall i\Gamma_i$

proof: By R0 it suffices to prove $(ss(DL\ \kappa\ \forall i\Gamma_i\ \alpha_i{:}\beta_{ij}/\chi_i))\rightarrow\forall i\Gamma_i$ which is equivalent to:

$(ss(DL\ \kappa\ \forall i\Gamma_i\ \alpha_i{:}\beta_{ij}/\chi_i))\rightarrow\Gamma_i$. Unfolding ss gives: $(\forall s(([(DL\ \kappa\ \forall i\Gamma_i\ \alpha_i{:}\beta_{ij}/\chi_i)](mg\ s))\rightarrow(mg\ s)))\rightarrow\Gamma_i$ which by the meaning laws M0-M7 is equivalent to: $(\forall s(([(DL\ \kappa\ \forall i\Gamma_i\ \alpha_i{:}\beta_{ij}/\chi_i)](mg\ s))\rightarrow(mg\ s)))\rightarrow(mg\ '\Gamma_i)$. Backchaining on $(mg\ \ '\Gamma_i)$ with s in the hypothesis assigned to be $'\Gamma_i$ in the conclusion shows that it suffices to prove:

$([(DL\ \kappa\ \forall i\Gamma_i\ \alpha_i{:}\beta_{ij}/\chi_i)](mg\ '\Gamma_i))$ which by the meaning laws: M0-M7 is equivalent to: $([(DL\ \kappa\ \forall i\Gamma_i\ \alpha_i{:}\beta_{ij}/\chi_i)]\Gamma_i)$ which by the laws of S5 is equivalent to: $([(DL\ \kappa\ \forall i\Gamma_i\ \alpha_i{:}\beta_{ij}/\chi_i)]\forall i\Gamma_i)$ which is an instance of MD1. QED.

MD4: $(([ss(DL\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)]\alpha_i)\wedge(\wedge_{j=1,mi}<\kappa>\beta_{ij}))\rightarrow ([ss(DL\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)]\chi_i)$

proof: Unfolding the last ss gives:

$(([ss(DL\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)]\alpha_i)\wedge\wedge_{j=1,mi}(<\kappa>\beta_{ij}))\rightarrow([\forall s(([(DL\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)](mg\ s))\rightarrow(mg\ s))]\chi_i)$

Instantiating s in the hypothesis to $'\chi_i$ and then dropping the hypothesis gives:

$(([ss(DL\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)]\alpha_i)\wedge\wedge_{j=1,mi}(<\kappa>\beta_{ij}))\rightarrow([((([(DL\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)](mg\ '\chi_i))\rightarrow(mg\ '\chi_i))]\chi_i)$. Using the meaning laws M0-M7 gives: $(([ss(DL\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)]\alpha_i)\wedge\wedge_{j=1,mi}(<\kappa>\beta_{ij}))\rightarrow([((([(DL\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)]\chi_i)\rightarrow\chi_i)]\chi_i)$.Backchaining on χ_i, it suffices to prove: $(([ss(DL\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)]\alpha_i)\wedge\wedge_{j=1,mi}(<\kappa>\beta_{ij}))\rightarrow([(DL\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)]\chi_i)$

By SS1 and the first hypothesis it suffices to prove:

$(([(DL\kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)]\alpha_i)\wedge\wedge_{j=1,mi}(<\kappa>\beta_{ij}))\rightarrow([(DL\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)]\chi_i)$ which is an instance of MD2. QED.

Finally MD5, MD6, and MD7 show that talking about the meanings of sets of FOL sentences in the modal representation of Default Logic is equivalent to talking about propositions in general.

MD5: $(\exists p((ms\ p)\wedge([(ms\ p)](\forall i\Gamma_i))\wedge\forall_i((([(ms\ p)]\alpha_i)\wedge\wedge_{j=1,mi}<\kappa>\beta_{ij})\rightarrow[(ms\ p)]\chi_i)))\equiv(DL\ \kappa(\forall i\Gamma_i)\alpha_i{:}\beta_{ij}/\chi_i)$

proof: The proof divides into two entailments:

(1) $[\exists p((ms\ p)\wedge([(ms\ p)]\forall i\Gamma_i)\wedge\forall_i((([(ms\ p)]\alpha_i)\wedge\wedge_{j=1,mi}(<\kappa>\beta_{ij}))\rightarrow[(ms\ p)]\chi_i))]\ (DL\ \kappa(\forall i\Gamma_i)\alpha_i{:}\beta_{ij}/\chi_i)$

DL is unfolded giving: $[((ms\ p)\wedge([(ms\ p)](\forall i\Gamma_i))\wedge\forall_i((([(ms\ p)]\alpha_i)\wedge\wedge_{j=1,mi}<\kappa>\beta_{ij})\rightarrow[(ms\ p)]\chi_i))]$

$\exists p(p\wedge([p]\forall i\Gamma_i)\wedge\forall_i((([p]\alpha_i)\wedge\wedge_{j=1,mi}<\kappa>\beta_{ij})\rightarrow[p]\chi_i))$

Instantiating the quantified p in the conclusion to be (ms p) produces a tautology.

(2) $[(DL\ \kappa(\forall i\Gamma_i)\alpha_i{:}\beta_{ij}/\chi_i)]\ \exists p((ms\ p)\wedge([(ms\ p)]\forall i\Gamma_i)\wedge\forall_i((([(ms\ p)]\alpha_i)\wedge\wedge_{j=1,mi}<\kappa>\beta_{ij})\rightarrow[(ms\ p)]\chi_i))$

p is assigned to be the set: $\{s{:}\ [(DL\ \kappa(\forall i\Gamma_i)\alpha_i{:}\beta_{ij}/\chi_i)](mg\ s)\}$.

Since p only occurs in (ms p) and since $(ms\{s{:}\ ([(DL\ \kappa)](mg\ s))\})$ is equivalent to $(ss(DL\ \kappa))$ we get:

$[(DL\ \kappa)]((ss(DL\ \kappa))\wedge([(ss(DL\ \kappa))]\forall i\Gamma_i)\wedge\forall_i((([(ss(DL\ \kappa))]\alpha_i)\wedge\wedge_{j=1,mi}<\kappa>\beta_{ij})\rightarrow([(ss(DL\ \kappa))]\chi_i)))$

which holds by theorems SS1, MD3, and MD4. QED.

MD6: $(ss(DL\ \kappa(\forall i\Gamma_i)\alpha_i{:}\beta_{ij}/\chi_i))\equiv(DL\ \kappa(\forall i\Gamma_i)\alpha_i{:}\beta_{ij}/\chi_i)$

proof: In view of SS1, it suffices to prove: $([(ss(DL\ \kappa))](DL\ \kappa))$. Unfolding the second occurrence of DL gives:

$[(ss(DL\ \kappa))]\exists p(p\wedge([p](\forall i\Gamma_i))\wedge\forall_i((([p]\alpha_i)\wedge\wedge_{j=1,mi}<\kappa>\beta_{ij})\rightarrow[p]\chi_i))$. Letting p be $(ss(DL\ \kappa))$ then gives:

$[(ss(DL\ \kappa))]((ss(DL\ \kappa))\wedge([(ss(DL\ \kappa))](\forall i\Gamma_i))\wedge\forall_i((([(ss(DL\ \kappa))]\alpha_i)\wedge\wedge_{j=1,mi}<\kappa>\beta_{ij})\rightarrow([(ss(DL\ \kappa))]\chi_i)))$

which holds by theorems MD3 and MD4. QED.

MD7: $(\kappa\equiv(DL\ \kappa(\forall i\Gamma_i)\alpha_i{:}\beta_{ij}/\chi_i))\rightarrow\exists s(\kappa\equiv(ms\ s))$

proof: From the hypothesis and MD6 $\kappa\equiv(ss(DL\ \kappa))$ is derived. Using the hypothesis to replace $(DL\ \kappa)$ by κ in this result gives: $\kappa\equiv(ss(DL\ \kappa))$, By SS2 this implies the conclusion. QED.

6. Conclusion: The Relationship between Default Logic and the Modal Logic

The relationship between the proof theoretic definition of Default Logic [Reiter 1980] and the modal representation is proven in two steps. First theorem DL1 shows that the meaning of the set dl is the proposition DL and then theorem DL2 shows that a set of FOL sentences which contains its FOL theorems is a fixed-point of the fixed-point equation of Default Logic with an initial set of axioms and defaults if and only if the meaning (or rather disquotation) of that set of sentences is logically equivalent to DL of the meanings of that initial set of sentences and those defaults.

DL1: $(ms(dl(fol\ '\kappa)\{'\Gamma_i\}\ '\alpha_i{:}'\beta_{ij}/'\chi_i))\equiv(DL(ms\ '\kappa)(\forall i\Gamma_i)\alpha_i{:}\beta_{ij}/\chi_i)$

proof: $(ms(dl(fol\ '\kappa)\{'\Gamma_i\}\ '\alpha_i{:}'\beta_{ij}/'\chi_i))$ Unfolding the definition of dl gives:

$ms(\cap\{p{:}\ (p\supseteq(fol\ p))\wedge(p\supseteq\{'\Gamma_i\})\wedge\forall_i((('\alpha_i\varepsilon p)\wedge\wedge_{j=1,mi}('(\neg\beta_{ij})\notin(fol\ '\kappa)))\rightarrow('\chi_i\varepsilon p))\})$. By FOL6 this is:

$ms\{s{:}\forall p((((fol\ p)\supseteq\{'\Gamma_i\})\wedge\forall_i((('\alpha_i\varepsilon(fol\ p))\wedge\wedge_{j=1,mi}('(\neg\beta_{ij})\notin(fol\ '\kappa)))\ \rightarrow('\chi_i\varepsilon(fol\ p))))\ \rightarrow(s\varepsilon(fol\ p)))\}$

Using C1 four times, C3, and FOL4 this is equivalent to: $ms\{s{:}\forall p((([(ms\ p)](ms\{'\Gamma_i\}))\wedge\forall_i((([(ms\ p)](mg\ '\alpha_i))\wedge\wedge_{j=1,mi}\neg([(ms\ '\kappa)](mg\ '\neg\beta_{ij})))\ \rightarrow([(ms\ p)](mg\ '\chi_i))))\rightarrow([(ms\ p)](mg\ s)))\}$

By the meaning laws M0-M7 this is equivalent to:

$ms\{s{:}\forall p((([(ms\ p)](ms\{'\Gamma_i\}))\ \wedge\forall_i((([(ms\ p)]\alpha_i)\wedge\wedge_{j=1,mi}\neg([(ms\ '\kappa)]\neg\beta_{ij}))\rightarrow([(ms\ p)]\chi_i)))\ \ \rightarrow([(ms\ p)](mg\ s)))\}$

By MS2 this is equivalent to:

$ms\{s{:}\forall p((([(ms\ p)](\forall i\Gamma_i))\ \wedge\forall_i((([(ms\ p)]\alpha_i)\wedge\wedge_{j=1,mi}\neg([(ms\ '\kappa)]\neg\beta_{ij}))\rightarrow([(ms\ p)]\chi_i)))\ \ \rightarrow([(ms\ p)](mg\ s)))\}$

Folding <> gives: $ms\{s{:}\forall p((([(ms\ p)](\forall i\Gamma_i))\wedge\forall_i((([(ms\ p)]\alpha_i)\wedge\wedge_{j=1,mi}(<(ms\ '\kappa)>\beta_{ij}))\ \rightarrow([(ms\ p)]\chi_i)))\rightarrow([(ms\ p)](mg\ s)))\}$

By S5 Modal Quantificational Logic this is equivalent to:

$ms\{s{:}\ ([\exists p((ms\ p)\wedge([(ms\ p)](\forall i\Gamma_i))\wedge\forall_i((([(ms\ p)]\alpha_i)\wedge\wedge_{j=1,mi}(<(ms\ '\kappa)>\beta_{ij}))\rightarrow([(ms\ p)]\chi_i)))]\ (mg\ s))\}$

By MD5 this is equivalent to: $ms\{s{:}\ ([(DL(ms\ '\kappa)(\forall i\Gamma_i)\alpha_i{:}\beta_{ij}/\chi_i)](mg\ s))\}$

Unfolding ms and lambda conversion gives: $\forall s(([(DL(ms\ '\kappa)(\forall i\Gamma_i)\alpha_i{:}\beta_{ij}/\chi_i)](mg\ s))\rightarrow(mg\ s))$

Folding ss gives: $ss(DL(ms\ '\kappa)(\forall i\Gamma_i)\alpha_i{:}\beta_{ij}/\chi_i)$. By MD6 is equivalent to:$(DL(ms\ '\kappa)(\forall i\Gamma_i)\alpha_i{:}\beta_{ij}/\chi_i)$ QED.

DL2: $((fol\ '\kappa)=(dl(fol\ '\kappa)\{'\Gamma_i\}\ '\alpha_i{:}'\beta_{ij}/'\chi_i))\leftrightarrow((ms\ '\kappa)\equiv(DL(ms\ '\kappa)(\forall i\Gamma_i)\alpha_i{:}\beta_{ij}/\chi_i))$

proof: By FOL8 $(fol\ '\kappa)=(dl(fol\ '\kappa)\{'\Gamma_i\}\ '\alpha_i{:}'\beta_{ij}/'\chi_i)$ is equivalent to: $(fol\ '\kappa)=(fol(dl(fol\ '\kappa)\{'\Gamma_i\}\ '\alpha_i{:}'\beta_{ij}/'\chi_i))$.

By C4 this is equivalent to: $(ms\ '\kappa)\equiv(ms(dl(fol\ '\kappa)\{'\Gamma_i\}\ '\alpha_i{:}'\beta_{ij}/'\chi_i))$.

By DL1 this is equivalent to: $(ms\ '\kappa)\equiv(DL(ms\ '\kappa)(\forall i\Gamma_i)\alpha_i{:}\beta_{ij}/\chi_i)$ QED.

Theorem DL2 shows that the set of theorems: $(fol\ '\kappa)$ of a set $'\kappa$ is a fixed-point of a fixed-point equation of Default Logic if and only if the meaning $(ms\ '\kappa)$ of $'\kappa$ is a solution to the necessary equivalence. Furthermore, by FOL9 there are no other fixed-points (such as a set not containing all its theorems) and by MD7 there are no other solutions (such as a proposition not representable as a sentence in the FOL object language). Therefore, the Modal representation of Default Logic (i.e. DL), faithfully represents the set theoretic description of Default Logic (i.e. dl).Finally, we note that $(\forall i\Gamma_i)$ and $(ms\ '\kappa)$ may be generalized to be arbitrary propositions Γ and κ giving the more general modal representation: $\kappa\equiv(DL\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)$.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant Numbers: MIP-9526532, EIA-9818341, and EIA-9972843.

Bibliography

[Antoniou 1997] Antoniou, Grigoris 1997. *NonMonotonic Reasoning*, MIT Press.

[Boyer&Moore 1981] R. S. Boyer and J. Strother Moore, "Metafunctions: proving them correct and using them efficiently as new proof procedures," *The Correctness Problem in Computer Science,* R. S. Boyer and J. Strother Moore, eds., Academic Press, New York, 1981.

[Bressan 1972] Bressan, Aldo 1972. *A General Interpreted Modal Calculus,* Yale University Press.

[Brown 1978] Brown,F.M., "A Semantic Theory for Logic Programming", *Colloquia Mathematica Societatis Janos Bolyai 26*, *Mathematical Logic in Computer Science*, Salgotarjan, Hungry, 1978.

[Brown 1987] Brown, Frank M. 1987. "The Modal Logic Z", In *The Frame Problem in AI"; Proc. of the 1987 AAAI Workshop*, Morgan Kaufmann, Los Altos, CA .

[Brown 1989] Brown, Frank M. 1989. "The Modal Quantificational Logic Z Applied to the Frame Problem", advanced paper *First International Workshop on Human & Machine Cognition*, May 1989 Pensacola, Florida. Abbreviated version published in *International Journal of Expert Systems Research and Applications, Special Issue: The Frame Problem. Part A.* eds. Keneth Ford and Pattrick Hayes, vol. 3 number 3, pp169-206 JAI Press 1990. Reprinted in *Reasoning Agents in a Dynamic World: The Frame problem*, editors: Kenneth M. Ford, Patrick J. Hayes, JAI Press 1991.

[Carnap 1946] Carnap, Rudolf 1946. "Modalities and Quantification" *Journal of Symbolic Logic*, vol. 11, number 2.

[Carnap 1956] Carnap, Rudolf 1956. *Meaning and Necessity: A Study in the Semantics of Modal Logic*, The University of Chicago Press.

[Fine 1970] Fine, K. 1970. "Propositional Quantifiers in Modal Logic" *Theoria* 36, p336--346.

[Hendry & Pokriefka 1985] Hendry, Herbert E. and Pokriefka, M. L. 1985. "Carnapian Extensions of S5", *Journal of Phil. Logic 14*.

[Hughes & Cresswell 1968] Hughes,G. E. & Creswell,M. J., 1968. *An Introduction to Modal Logic*, Methuen & Co. Ltd., London.

[Lewis 1936] Lewis, C. I. 1936. Strict Implication, *Journal of Symbolic Logic*, vol I.

[Mendelson 1964] Mendelson, E. 1964. *Introduction to Mathematical Logic*, Van Norstrand, Reinhold Co., New York.

[Parks 1976] Parks, Z. 1976. "An Investigation into Quantified Modal Logic", Studia Logica 35, p109-125.

[Quine 1969] Quine, W.V.O., Set Theory and Its Logic, revised edition, Oxford University Press, London, 1969.

[Reiter 1980] Reiter, R. 1980. "A Logic for Default Reasoning" *Artificial Intelligence*, 13.

Author information

Frank M. Brown- Artificial Intelligence Laboratory, University of Kansas, Lawrence, Kansas, 66045, e-mail: brown@ku.edu.

ON THE RELATIONSHIP BETWEEN QUANTIFIED REFLECTIVE LOGIC AND QUANTIFIED DEFAULT LOGIC

Frank M. Brown

Abstract: Reflective Logic and Default Logic are both generalized so as to allow universally quantified variables to cross modal scopes whereby the Barcan formula and its converse hold. This is done by representing both the fixed-point equation for Reflective Logic and the fixed-point equation for Default both as necessary equivalences in the Modal Quantificational Logic Z. and then inserting universal quantifiers before the defaults. The two resulting systems, called Quantified Reflective Logic and Quantified Default Logic, are then compared by deriving metatheorems of Z that express their relationships. The main result is to show that every solution to the equivalence for Quantified Default Logic is a strongly grounded solution to the equivalence for Quantified Reflective Logic. It is further shown that Quantified Reflective Logic and Quantified Default Logic have exactly the same solutions when no default has an entailment condition.

Keywords: Quantified Reflective Logic, Quantified Default Logic, Modal Logic, Nonmonotonic Logic.

1. Introduction

Two nonmonotonic logics which inherently deal with entailment conditions in addition to possibility conditions in their defaults; are Reflective Logic and Default Logic [Reiter 1980] [Antoniou 1997]. The fixed-point solutions to Default Logic are defined by the set theoretic equation $\kappa=(\mathrm{dl}\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)$ where:

$$(\mathrm{dl}\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i) =\mathrm{df}\ \cap\{p{:}(p\supseteq(\mathrm{fol}\ p))\wedge(p\supseteq\Gamma)\wedge\ \wedge_i(((\alpha_i\varepsilon p)\wedge\ \wedge_{j=1,mi}((\neg\beta_{ij})\notin\kappa))\rightarrow(\chi_i\varepsilon p))\}$$

where α_i, β_{ij}, and χ_i are closed sentences of First Order Logic (i.e. FOL) and Γ is a set of closed sentences of FOL $\wedge_{j=1,mi}$ stands for the conjunction of the formula that follows it as j ranges from 1 to m_i. If $m_i=0$ then it specifies #t. $\wedge_i$ is also a conjunction. By closed it is meant that no sentence may contain a free variable. (fol p) is the set of theorems deducible in FOL from the set p. The fixed-point solutions for Reflective Logic, can be defined by the simpler set theoretic equation $\kappa=(\mathrm{rl}\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)$ given in [Brown 1989] where:

$$(\mathrm{rl}\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i) =\mathrm{df}\ \mathrm{fol}(\Gamma\cup\{\chi_i{:}(\alpha_i\varepsilon\kappa)\wedge\ \wedge_{j=1,mi}((\neg\beta_{ij})\notin\kappa)\})$$

where α_i, β_{ij}, and χ_i are again closed sentences of FOL and Γ is a set of closed sentences of FOL.

These two nonmonotonic systems have the basic problem that they do not explicate the case where free variables occur in the α_i, β_{ij}, and χ_i sentences and which are universally quantified just over the scope of those sentences. To carry out such an explication we want to transform $(\mathrm{dl}\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)$ into something like:

$$\cap\{p{:}(p\supseteq(\mathrm{fol}\ p))\wedge(p\supseteq\Gamma)\wedge\ \wedge_i\underline{\forall\boldsymbol{\xi}_{\mathbf{i}}}(((\alpha_i\varepsilon p)\wedge\ \wedge_{j=1,mi}((\neg\beta_{ij})\notin\kappa))\rightarrow(\chi_i\varepsilon p))\}$$

and $(\mathrm{rl}\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)$ into something like:[5] $\mathrm{fol}(\Gamma\cup\{\Psi{:}\ \vee_i\underline{\exists\boldsymbol{\xi}_{\mathbf{i}}}(\Psi=\chi_i\wedge(\alpha_i\varepsilon\kappa)\wedge\ \wedge_{j=1,mi}((\neg\beta_{ij})\notin\kappa))\})$
where $\underline{\xi}_i$ is a sequence of variables and the universal quantifier really means universal quantification. That is, the Barcan formula and its converse hold [Carnap 1946] so that a property universally holds (in κ) if and only if it holds (in κ) for everything: $((\underline{\forall\xi}\alpha)\varepsilon\kappa)\leftrightarrow(\underline{\forall\xi}(\alpha\varepsilon\kappa))$. The problem lies in the fact that α_i, β_{ij}, and χ_i are necessarily closed sentences of FOL.[6]

However, [Brown 2003a] showed how Reflective Logic can be represented in Modal Logic by the necessary equivalence: $\kappa\equiv(\mathrm{RL}\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)$ where:

$$(\mathrm{RL}\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i) =\mathrm{df}\ \Gamma\wedge\wedge_i((([\kappa]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow\chi_i)$$

Likewise [Brown 2003b] showed how Default Logic can be represented in Modal Logic by the necessary equivalence: $\kappa\equiv(\mathrm{DL}\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)$ where:

$$(\mathrm{DL}\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i) =\mathrm{df}\ \exists p(p\wedge([p]\Gamma)\wedge\wedge_i((([p]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow([p]\chi_i)))$$

[5] When the set theoretic notation is unravelled the existential quantifiers specified herein are essentially universally quantified over the defaults as can be seen in the equivalent equation: $k=\cap\{p{:}(p\supseteq(\mathrm{fol}\ p))\wedge(p\supseteq\Gamma)\wedge\ \wedge_i\underline{\forall\xi}_i(((\alpha_i\varepsilon k)\wedge\wedge_{j=1,mi}((\neg\beta_{ij})\notin k))\rightarrow(\chi_i\varepsilon p))\}$

[6] Of course one generally gives a meaning to such a sentence by saying that all the free variables are implicitly universally quantified or that all such variables are implicitly existentially quantified. However, neither approach allows a quantifier to refer to the same free variable in α_i, β_{ij}, and χ_i. This issue is discussed in more detail in section 3.2 in [Antoniou 1997].

The advantage of the modal representations is that quantifiers can be embedded in them wherever we wish thus allowing inserted universal quantifiers to capture the free variables in α_i, β_{ij}, and χ_i , giving the generalizations:

$(\mathrm{QRL}\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i) =\mathrm{df}\ \Gamma\wedge\wedge_i\underline{\forall\xi_i}((([\kappa]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow\chi_i)$

$(\mathrm{QDL}\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i) =\mathrm{df}\ \exists p(p\wedge([p]\Gamma)\wedge\wedge_i\underline{\forall\xi_i}((([p]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow([p]\chi_i)))$

Having created two new nonmonotonic systems (i.e. QRL and QDL) the question arises as to how their fixed-point solutions are related. Herein we address this question. Section 2 axiomatizes the Z Modal Quantificational Logic. Quantified Reflective Logic (i.e., QRL) is defined in section 3 and some basic theorem schemata about it are proven. Quantified Default Logic (i.e., QDL) is defined in section 4 and some basic theorem schemata about it are proven. The main result is proven in section 5. Finally, some conclusions are drawn in section 6.

2. Axiomatization of Z Modal Logic

The Modal Quantificational Logic Z [Brown 1987] is a seven tuple: (→, #f, ∀, [], *vars*, *predicates*, *functions*) where →, #f, ∀, and [] are logical symbols, *vars* is a set of variable symbols, *predicates* is a set of predicate symbols each of which has an implicit arity specifying the number of terms associated with that predicate, and *functions* is a set of function symbols each of which has an implicit arity specifying the number of terms associated with that function. The sets of logical symbols, variables, predicate symbols, and function symbols are pairwise disjoint. The set of terms is the smallest set which includes the variables and is closed under the process of forming new terms from other terms using the function symbols of the language. The set of sentences is the smallest set which includes #f, the *variables*, and each of the *predicates* followed by an appropriate number of terms, and is closed under the process of forming new sentences from other sentences using the logical symbols of the language, provided that no variable in any subexpression has free occurrences both as a sentence and as a term. Variables that occur only in term positions are called concept variables. Variables which occur only in sentence positions are called propositional variables. Lower case Roman letters possibly indexed with digits are used as variables of Z. Greek letters are used as syntactic metavariables. γ, $\gamma_1,\ldots\gamma_n$, range over the variables, ξ, $\xi_1,\ldots\xi_n$ range over a sequence of variables of an appropriate arity, $\pi,\pi_1\ldots\pi_n,\rho,\rho_1\ldots\rho_n$ range over the predicate symbols, ϕ, $\phi_1\ldots\phi_n$ range over function symbols, δ, $\delta_1\ldots\delta_n$ range over terms, Δ, $\Delta_1\ldots\Delta_n$ range over a sequence of terms of an appropriate arity, and α, $\alpha_1\ldots\alpha_n$, β, $\beta_1\ldots\beta_n$,χ, $\chi_1\ldots\chi_n$, Γ, and Ψ range over sentences. Thus, the terms are of the forms γ and (ϕ $\delta1\ldots\delta n$), and the sentences are of the forms ($\alpha\rightarrow\beta$), #f, ($\forall\gamma\ \alpha$), ([]α), ($\pi\ \delta_1\ldots\delta_n$), and γ. A nullary predicate π or function ϕ is written as a sentence or term without parentheses. The primitive symbols of Z are shown in Figure 1.

Symbol	Meaning		Symbol	Meaning
$\alpha\rightarrow\beta$	if α then β.		$\forall\gamma\ \alpha$	for all γ, α.
#f	falsity		[] α	α is logically necessary

Figure 1: Primitive Symbols of Z

The defined symbols of Z are listed in Figure 2 below with their intuitive interpretations.

Symbol	Definition	Meaning		Symbol	Definition	Meaning
$\neg\alpha$	$\alpha\rightarrow$ #f	not α		$\alpha\wedge\beta$	$\neg(\alpha\rightarrow\neg\beta)$	α and β
#t	$\neg$ #f	truth		$\alpha\leftrightarrow\beta$	$(\alpha\rightarrow\beta)\wedge(\beta\rightarrow\alpha)$	α if and only if β
$\alpha\vee\beta$	$(\neg\alpha)\rightarrow\beta$	α or β		$\exists\gamma\ \alpha$	$\neg\forall\gamma\ \neg\alpha$	for some γ , α
<> α	$\neg[]\neg\alpha$	α is logically possible		$[\beta]\ \alpha$	$([](\beta\rightarrow\alpha))$	β entails α
$\alpha\equiv\beta$	$[](\alpha\leftrightarrow\beta)$	α is logically equivalent toβ		$<\beta>\ \alpha$	$(<>(\beta\wedge\alpha))$	α is possible with β
$\delta_1=\delta_2$	$(\pi\delta_1)\equiv(\pi\ \delta_1)$	$\delta_1$is logically equal to δ_2				

Figure 2: Defined Symbols of Z

Z is effectively axiomatized with a recursively enumerable set of theorems as the set of axioms is itself recursively enumerable and its inference rules are recursive. The classical (i.e., non-modal) axioms and inference rules of Z include those of Quantificational Logic [Mendelson 1964] given in Figure 3. The laws MR1, MR2, MA1-MA7 are a standard set of axioms and inference rules for First Order Quantificational Logic except for the following: point: Because γ in MR2, MA4, and MA5 may be a propositional variable these laws constitute a fragment of Second Order Logic. Propositional quantifiers in modal logics have been investigated in [Fine 1970].

MA1: α → (β→ α) MR1: from α and (α→ β) infer β
MA2: (α→ (β→ ρ)) → ((α→ β)→ (α→ ρ)) MR2: from α infer (∀γ α)
MA3: ((¬ α)→ (¬ β))→ (((¬ α)→ β)→α)
MA4: (∀γ α)→ β where β is the result of substituting an expression (which is free for the free positions of γ in α) for all the free occurrences of γ in α.
MA5: ((∀γ (α → β)) → (α →(∀γ β))) where γ does not occur in α.

Figure 3: The Classical Rules and Axioms of Z

The modal inference rule and axioms of Z about logical necessity (i.e., []) are given in Figure 4. R0, A1, A2, and A3 constitute an S5 Modal Logic [Hughes and Cresswell 1968] which, with the nonmodal laws, is an S5 modal quantificational logic similar to [Carnap 1946], [Carnap 1956], and a First Order Logic version [Parks 1976] of [Bressan 1972] in which the Barcan formula: (∀γ([]α))→([]∀γα) and its converse hold. R0 implies that all assertions are logically necessary. Thus, in any logic with R0, contingent facts Γ holding in a knowledgebase κ are specified by asserting ([κ]Γ). If Γ is all that is in κ then κ≡Γ is asserted. The variable κ may occur in Γ.

R0: from α infer ([] α) A4: ([]α) → ([](α{π / λξβ}
A1: ([]p) → p A5: ([]α) → ([](α{φ / λξδ}
A2: ([p]q) → (([]p)→ ([]q)) A6: ¬(∀x∀y(x=y))
A3: ([]p) ∨ ([]¬[]p))

Figure 4: The Modal Inference Rule and Axioms of Z

A4 is the key axiom schema of Z. It is far stronger than the trivial possibility axioms such as ∃pq((¬[p]q)∧(¬[p]¬q)) assumed in [Lewis 1936] and ∃p((<>p)∧(<>¬p)) assumed in [Bressan 1972]. It also extends certain axiom schemata used in propositional logic, including the PropPosAx schema in [Brown 1979], S13 [Cocchiarella 1984], and S5c [Hendry and Pokriefka 1985].

3. Quantified Reflective Logic

The formula for Quantified Reflective Logic[7] (i.e., QRL) [Brown 1989] is defined in Z as follows:

RL0: $(\mathrm{QRL}\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)$ =df $\Gamma\wedge\wedge_i\forall\xi_i((([\kappa]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow\chi_i)$

where Γ, α_i, β_{ij}, and χ_i are sentences of Z and κ does not occur in ξ. These sentences may contain free variables some of which may be captured by the $\forall\xi_i$ quantifiers. When the context is obvious Γ $\alpha_i{:}\beta_{ij}/\chi_i$ is omitted and instead just (QRL κ) is written. Interpreted as a doxastic logic, the equivalence:

$$\kappa\equiv(\mathrm{QRL}\ \kappa)$$

states:

[7] In the QRL generalization of Reflective Logic the Barcan formula and its converse hold for [k]: ([k]∀ξα)↔(∀ξ[k]α) since they are inherited from the S5 modal properties of []. In terms of set theoretic fixed-points this amounts to saying that ('($\underline{\forall\xi}$α)εk)→($\underline{\forall\xi}$('αεk)) holds except for the problem that that in the set theory representation 'α is a closed sentence.

that which is believed is logically equivalent to
Γ and for each i, for all ξ_i if α_i is believed and for each j, β_{ij} is believable then χ_i

Here are some simple properties of QRL, namely that (QRL κ) entails Γ and any conclusion χ_i of a default whose conditions hold:

R1: $[(\mathrm{QRL}\ \kappa)]\Gamma$

proof: Unfolding QRL gives: $[\Gamma\wedge\wedge_i\forall\xi_i((([\kappa]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow\chi_i)]\Gamma$ which is a tautology. QED.

R2: $(([\kappa]\alpha_i)\wedge(\wedge_{j=1,m_i}(<\kappa>\beta_{ij})))\rightarrow([(\mathrm{QRL}\ \kappa)]\chi_i)$

proof: Unfolding QRL gives: $(([\kappa]\alpha_i)\wedge(\wedge_{j=1,m_i}(<\kappa>\beta_{ij})))\rightarrow([\Gamma\wedge\wedge_i\forall\xi_i((([\kappa]\alpha_i)\wedge(\wedge_{j=1,m_i}(<\kappa>\beta_{ij})))\rightarrow\chi_i)]\chi_i)$

Using the hypotheses on the ith instance and where the quantified ξ_i is instantiated to ξ_i gives:

$(([\kappa]\alpha_i)\wedge(\wedge_{j=1,m_i}(<\kappa>\beta_{ij})))\rightarrow([\Gamma\wedge\wedge_i\forall\xi_i((([\kappa]\alpha_i)\wedge(\wedge_{j=1,m_i}(<\kappa>\beta_{ij})))\rightarrow\chi_i)\wedge\chi_i]\chi_i)$ which is a tautology. QED.

4. Quantified Default Logic

The formula for Quantified Default Logic (i.e., QDL) [Brown 1989] is defined in Z as follows:

D0: $(\mathrm{QDL}\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i) =_{\mathrm{df}} \exists p(p\wedge([p]\Gamma)\wedge\wedge_i\forall\xi_i((([p]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow([p]\chi_i)))$

where Γ, α_i, β_{ij}, and χ_i are sentences of Z without any free occurences of p and neither p nor κ occur in ξ_i. These sentences may contain free variables some of which may be captured by the $\forall\xi_i$ quantifiers. When the context is obvious $\Gamma\ \alpha_i{:}\beta_{ij}/\chi_i$ is omitted and just (QDL κ) is written. Interpreted as a doxastic logic the equivalence:

$$\kappa\equiv(\mathrm{QDL}\ \kappa)$$

states:

that which is believed is logically equivalent to
the disjunction of all potential belief states such that:
Γ is potentially believed
and for each i, for all ξ
if α_i is potentially believed and for each j, β_{ij} is believable then χ_i is potentially believed.

Given below are some simple properties of QDL. The first two state that QDL entails Γ and any conclusion χ_i of a default whose entailment condition holds in QDL and whose possible conditions are possible with κ.

D1: $[(\mathrm{QDL}\ \kappa)]\Gamma$

proof: Unfolding QDL gives: $[\exists p(p\wedge([p]\Gamma)\wedge\wedge_i\forall\xi_i((([p]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow([p]\chi_i)))]\Gamma$

Since p is not free in Γ, pulling $\exists$p out of the hypothesis of the entailment gives:

$\forall p((([p]\Gamma)\wedge\wedge_i\forall\xi_i((([p]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow([p]\chi_i)))\rightarrow([p]\Gamma))$ which is a tautology. QED.

D2: $(([(\mathrm{QDL}\ \kappa)]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow([(\mathrm{QDL}\ \kappa)]\chi_i)$

proof: Unfolding both occurrences of QDL gives:

$(([\exists p(p\wedge([p]\Gamma)\wedge\wedge_i\forall\xi_i((([p]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow([p]\chi_i)))]\alpha_i)\wedge(\wedge_{j=1,m_i}(<\kappa>\beta_{ij})))$
$\rightarrow([\exists p(p\wedge([p]\Gamma)\wedge\wedge_i\forall\xi_i((([p]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow([p]\chi_i)))]\chi_i)$

Since p is not free in α_i and χ_i, pulling $\exists$p out of the hypotheses of the entailments gives:

$((\forall p((([p]\Gamma)\wedge\wedge_i\forall\xi_i((([p]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow([p]\chi_i)))\rightarrow([p]\alpha_i))\wedge(\wedge_{j=1,m_i}(<\kappa>\beta_{ij}))))$
$\rightarrow\forall p((([p]\Gamma)\ \wedge\wedge_i\forall\xi_i((([p]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow([p]\chi_i)))\rightarrow([p]\chi_i))$

Instantiating the p in the hypothesis to the p in the conclusion gives:

$(((([p]\Gamma)\wedge\wedge_i\forall\xi_i((([p]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow([p]\chi_i)))\rightarrow([p]\alpha_i))$
$\wedge(\wedge_{j=1,m_i}(<\kappa>\beta_{ij}))\wedge([p]\Gamma)\wedge\wedge_i\forall\xi_i((([p]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow([p]\chi_i)))\rightarrow([p]\chi_i)$

which simplifies to just:$(([p]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij}))\wedge([p]\Gamma)\wedge\wedge_i\forall\xi_i((([p]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow([p]\chi_i)))\rightarrow([p]\chi_i)$

Since p is not in ξ, forward chaining using the first and second hypotheses on the fourth proves the theorem. QED.

A slightly stronger version of QDL is defined below:
D3: $(\mathrm{QDL}^*\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)$ =df $\exists p(p\wedge([\kappa]p)\wedge([p]\Gamma)\wedge\wedge_i\forall\xi_i((([p]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow([p]\chi_i)))$

D4: [(QDL* κ)](QDL κ)
proof: Unfolding QDL* and QDL gives: $[\exists p(p\wedge([\kappa]p)\wedge([p]\Gamma)\ \wedge\wedge_i\forall\xi_i((([p]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow([p]\chi_i)))]$
$\exists p(p\wedge([p]\Gamma)\wedge\wedge_i\forall\xi_i((([p]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow([p]\chi_i)))$
Letting p in the conclusion be the p in the hypothesis results in a tautology. QED.

Theorem D5 shows that QDL and QDL* are logically equivalent whenever κ entails the QDL formula:
D5: $([\kappa](\mathrm{QDL}\ \kappa))\rightarrow((\mathrm{QDL}\ \kappa)\equiv(\mathrm{QDL}^*\ \kappa))$
proof: From Theorem D4, it suffices to prove: $([\kappa](\mathrm{QDL}\ \kappa))\rightarrow([(\mathrm{QDL}\ \kappa)](\mathrm{QDL}^*\ \kappa))$
Unfolding QDL* gives:$([\kappa](\mathrm{QDL}\ \kappa))\rightarrow([(\mathrm{QDL}\ \kappa)]\exists p(p\wedge([\kappa]p)\wedge([p]\Gamma)\wedge\wedge_i\forall\xi_i((([p]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow([p]\chi_i))))$
Since p and κ are not in ξ and p is not free in Γ, α_i, β_{ij}, and χ_i, letting p be (QDL κ) gives:
$([\kappa](\mathrm{QDL}\ \kappa))\rightarrow$
$([(\mathrm{QDL}\ \kappa)]((\mathrm{QDL}\ \kappa)\wedge([\kappa](\mathrm{QDL}\ \kappa))\wedge([(\mathrm{QDL}\ \kappa)]\Gamma)\wedge\wedge_i\forall\xi_i((([(\mathrm{QDL}\ \kappa)]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow([(\mathrm{QDL}\ \kappa)]\chi_i))))$
which holds by D1, D2, and the hypothesis. QED

5. Relationship between QRL and QDL

The following theorems characterize the relationship between QDL and QRL:

RD1: $(\kappa\equiv(\mathrm{QDL}\ \kappa))\rightarrow[\kappa](\mathrm{QRL}\ \kappa)$
proof: Unfolding the definition of QRL gives: $(\kappa\equiv(\mathrm{QDL}\ \kappa))\rightarrow[\kappa](\Gamma\wedge\wedge_i\forall\xi_i((([\kappa]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow\chi_i))$
Since κ is not in ξ, pushing $[\kappa]$ to lowest scope using the laws of KU45 modal logic on $[\kappa]$ gives:
$(\kappa\equiv(\mathrm{QDL}\ \kappa))\rightarrow((([\kappa]\Gamma)\wedge\wedge_i\forall\xi_i((([\kappa]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow([\kappa]\chi_i)))$
Since κ is not in ξ, using the hypothesis to replace the first κ in the conclusion by (QDL κ) gives $[(\mathrm{QDL}\ \kappa)]\Gamma$ which by theorem D1 is true. It remains only to prove: $(\kappa\equiv(\mathrm{QDL}\ \kappa))\rightarrow((([\kappa]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow([\kappa]\chi_i))$
Since κ is not in ξ_i, replacing two occurrences of κ by using the hypothesis and then dropping the hypothesis gives: $(([(\mathrm{QDL}\ \kappa)]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow([(\mathrm{QDL}\ \kappa)]\chi_i)$ which by theorem D2 is true. QED.

RD2: $(\kappa\equiv(\mathrm{QDL}\ \kappa))\rightarrow[(\mathrm{QRL}\ \kappa)]\kappa$
proof: Using the hypothesis to replace the entailed κ in the conclusion gives: $(\kappa\equiv(\mathrm{QDL}\ \kappa))\rightarrow([(\mathrm{QRL}\ \kappa)](\mathrm{QDL}\ \kappa))$
Unfolding QDL in the conclusion gives:
$(\kappa\equiv(\mathrm{QDL}\ \kappa))\rightarrow[(\mathrm{QRL}\ \kappa)]\exists p(p\wedge([p]\Gamma)\ \wedge\wedge_i\forall\xi_i\ ((([p]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow([p]\chi_i)))$
Since p and κ are not in ξ and p is not free in Γ, α_i, β_{ij}, and χ_i, letting p be (QRL κ) gives:
$(\kappa\equiv(\mathrm{QDL}\ \kappa))\rightarrow[(\mathrm{QRL}\ \kappa)]((\mathrm{QRL}\ \kappa)\wedge([(\mathrm{QRL}\ \kappa)]\Gamma)\wedge\wedge_i\forall\xi_i((([(\mathrm{QRL}\ \kappa)]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow([(\mathrm{QRL}\ \kappa)]\chi_i)))$
The hypothesis $\kappa\equiv(\mathrm{QDL}\ \kappa)$ and RD1 imply $([\kappa](\mathrm{QRL}\ \kappa))$ which, since κ is not in ξ, allows the above sentence to be generalized to:
$(\kappa\equiv(\mathrm{QDL}\ \kappa))\rightarrow[(\mathrm{QRL}\ \kappa)]((\mathrm{QRL}\ \kappa)\wedge([(\mathrm{QRL}\ \kappa)]\Gamma)\wedge\wedge_i\forall\xi_i((([\kappa]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow([(\mathrm{QRL}\ \kappa)]\chi_i)))$
which by RL1 and RL2 is true. QED.

From RD1 and RD2 we may infer that every solution of the reflective equivalence of Quantified Default Logic is a solution of the equivalence for Quantified Reflective Logic:
RD3: $(\kappa\equiv(\mathrm{QDL}\ \kappa))\rightarrow (\kappa\equiv(\mathrm{QRL}\ \kappa))$

It also follows that every solution to Quantified Reflective Logic entails (QDL κ).
RD4: $([\kappa](\mathrm{QRL}\ \kappa))\rightarrow [\kappa](\mathrm{QDL}\ \kappa)$
proof: Unfolding the definition of QDL gives:
$([\kappa](\mathrm{QRL}\ \kappa))\rightarrow [\kappa]\exists p(p\wedge([p]\Gamma)\wedge\wedge_i\forall\xi_i((([p]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow([p]\chi_i)))$
Since p and κ are not in ξ_i and p is not free in Γ, α_i, β_{ij}, and χ_i, letting p be κ gives:
$([\kappa](\mathrm{QRL}\ \kappa))\rightarrow [\kappa](\kappa\wedge([\kappa]\Gamma)\wedge\wedge_i\forall\xi_i((([\kappa]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow([\kappa]\chi_i)))$
Since κ is not in ξ_i, using the hypothesis to replace two occurrences of κ by (QRL κ) gives the generalization:
$([\kappa](\mathrm{QRL}\ \kappa))\rightarrow [\kappa](\kappa\wedge([(\mathrm{QRL}\ \kappa)]\Gamma)\wedge\wedge_i\forall\xi_i((([\kappa]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow([(\mathrm{QRL}\ \kappa)]\chi_i)))$
which is true by RL1 and RL2. QED
.
From RD3 and RD4 we may infer that the solutions to QDL are precisely those solutions to QRL which are entailed by (QDL κ):
RD5: $(\kappa\equiv(\mathrm{QDL}\ \kappa)) \leftrightarrow ((\kappa\equiv(\mathrm{QRL}\ \kappa))\wedge([(\mathrm{QDL}\ \kappa)]\kappa))$

Likewise since $\kappa\equiv(\mathrm{QRL}\ \kappa)$ in RD5 implies $([\kappa](\mathrm{QDL}\ \kappa))$ by RD3 and since $([\kappa](\mathrm{QDL}\ \kappa))$ implies that (QDL κ) is logically equivalent to (QDL* κ) by D5, it follows that:
RD6: $(\kappa\equiv(\mathrm{QDL}\ \kappa)) \leftrightarrow ((\kappa\equiv(\mathrm{QRL}\ \kappa))\wedge([(\mathrm{QDL}^*\ \kappa)]\kappa))$

RD6 characterizes the relationship between QRL and QDL in terms of $[(\mathrm{QDL}^*\ \kappa)]\kappa$. We now show that $([(\mathrm{QDL}^*\ \kappa)]\kappa)$ is equivalent to the notion of being constructive, defined as follows: a Reflectivec solution κ is constructive iff it is not the case that there exists a proposition which satisfies the following four conditions: (1) κ entails that proposition, (2) the proposition does not entail κ, (3) the proposition entails Γ, and (4) for each i and for all ξ the proposition entails the conclusion χ_i of each default whose presupposition α_i is entailed by that proposition and whose β_{ij} formulas are possible with κ.
RD7: $(\mathrm{Constructive}\ \kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i)$
$=_{df}\neg\exists p(([\kappa]p)\wedge(\neg([p]\kappa))\wedge([p]\Gamma)\wedge\wedge_i\forall\xi((([p]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow([p]\chi_i)))$

RD8: $([(\mathrm{QDL}^*\ \kappa)]\kappa)\leftrightarrow(\mathrm{Constructive}\ \kappa)$
proof: Unfolding the (QDL* κ) in $([(\mathrm{QDL}^*\ \kappa)]\kappa)$ gives:
$[\exists p(p\wedge([p]\Gamma)\wedge([\kappa]p)\wedge\wedge_i\forall\xi_i((([p]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow([p]\chi_i)))]\kappa$
Pulling $\exists$p out of the hypothesis of the entailment gives:
$\forall p((\wedge_i\forall\xi_i((([p]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow([p]\chi_i))\wedge([p]\Gamma)\wedge([\kappa]p))\rightarrow([p]\kappa))$
Pushing a negation through the formula gives:
$\neg\exists p(([\kappa]p)\wedge(\neg([p]\kappa))\wedge([p]\Gamma)\ \wedge\wedge_i\forall\xi_i((([p]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow([p]\chi_i)))$
which is the definition of being constructive. QED.

Being constructive isequivalent to the notion of being strongly grounded. A Quantified Reflective solution κ is strongly grounded iff it is not the case that there exists a proposition which satisfies the following four conditions: (1) κ entails that proposition, (2) the proposition does not entail κ, (3) the proposition entails Γ, and (4) for each i and all ξ the proposition entails the conclusion χ_i of each default whose β_{ij} formulas are also possible with κ in addition to being such that the default's presupposition α_i is entailed by that proposition and the default's β_{ij} formulas are possible with that proposition:[8]

[8] When no variables cross modal scopesthis concept may be defined in set theory as:

RD9: (Strongly-grounded $\kappa\ \Gamma\ \alpha_i{:}\beta_{ij}/\chi_i$) =df

$\neg\exists p(([\kappa]p)\wedge(\neg([p]\kappa))\wedge([p]\Gamma)\ \wedge\wedge_i\forall\xi((\wedge_{j=1,mi}(<\kappa>\beta_{ij}))\rightarrow((([p]\alpha_i)\wedge(\wedge_{j=1,mi}(<p>\beta_{ij})))\rightarrow([p]\chi_i))))$

RD10: (Constructive κ)$\leftrightarrow$(Strongly-grounded κ)
proof: Unfolding Strongly-grounded gives:
$\neg\exists p(([\kappa]p)\wedge(\neg([p]\kappa))\wedge([p]\Gamma)\wedge\wedge_i\forall\xi_i((\wedge_{j=1,mi}(<\kappa>\beta_{ij}))\rightarrow((([p]\alpha_i)\wedge(\wedge_{j=1,mi}(<p>\beta_{ij})))\rightarrow([p]\chi_i))))$
Since ($[\kappa]p$), ($<\kappa>\beta_{ij}$) implies ($<p>\beta_{ij}$)). Since p and κ do not occur in ξ, the above sentence is equivalent to:
$\neg\exists p(([\kappa]p)\wedge(\neg([p]\kappa))\wedge([p]\Gamma)\wedge\wedge_i\forall\xi_i((\wedge_{j=1,mi}(<\kappa>\beta_{ij}))\rightarrow((([p]\alpha_i)\wedge\#t)\rightarrow([p]\chi_i))))$
or rather: $\neg\exists p(([\kappa]p)\wedge(\neg([p]\kappa))\wedge([p]\Gamma)\wedge\wedge_i\forall\xi_i((([p]\alpha_i)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow([p]\chi_i)))$
which is the definition of being constructive. QED.

The above theorems give five characterizations of QDL in terms of QRL:[9]
RD11: All the following are equivalent:
(1) $\kappa\equiv$(QDL κ), (2) ($\kappa\equiv$(QRL κ))$\wedge$($\kappa\equiv$(QDL κ)), (3) ($\kappa\equiv$(QRL κ))$\wedge$([(QDL κ)]κ), (4) ($\kappa\equiv$(QRL κ))$\wedge$([(QDL* κ)]κ),
(5) ($\kappa\equiv$(QRL κ))$\wedge$(Constructive κ), (6) ($\kappa\equiv$(QRL κ))$\wedge$(Strongly-grounded κ)
proof: The second formula follows from RD3, the third from RD5, the fourth from RD6, the fifth from RD8 and the sixth from RD10. QED.

Having shown that the Quantified Default solutions are the strongly grounded Quantified Reflective solutions, it is now shown that being strongly grounded essentially applies only to the defaults with entailment conditions since if there are essentially no entailment conditions in the defaults (i.e., α_i is #t for every ith default since #t is entailed by anything) then the Quantified Default solutions are precisely the Quantified Reflective solutions:
RD12: (QDL $\kappa\ \Gamma$ #t:β_{ij}/χ_i) $\equiv$ (QRL $\kappa\ \Gamma$ #t:β_{ij}/χ_i)
proof: Unfolding (QDL $\kappa\ \Gamma$ #t:β_{ij}/χ_i) gives: $\exists p(p\wedge([p]\Gamma)\wedge\wedge_i\forall\xi_i((([p]\#t)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow([p]\chi_i)))$
which simplifies to: $\exists p(p\wedge([p]\Gamma)\wedge\wedge_i\forall\xi_i((\wedge_{j=1,mi}(<\kappa>\beta_{ij}))\rightarrow([p]\chi_i)))$
Since p does not occur in ξ_i, the KU45 modal laws of [p] allow it to be pulled out giving:
$\exists p(p\wedge([p]\ (\Gamma\wedge\wedge_i\forall\xi((\wedge_{j=1,m_i}(<\kappa>\beta_{ij}))\rightarrow\chi_i))))$ which is: $\Gamma\wedge\wedge_i\forall\xi_i((\wedge_{j=1,mi}(<\kappa>\beta_{ij}))\rightarrow\chi_i)$
which may be rewritten as: $\Gamma\wedge\wedge_i\forall\xi_i((([\kappa]\#t)\wedge(\wedge_{j=1,mi}(<\kappa>\beta_{ij})))\rightarrow\chi_i)$ which is: (QRL $\kappa\ \Gamma$ #t:β_{ij}/χ_i). QED.

6. Conclusion

Theorem RD11 shows that the solutions to Quantified Default Logic (i.e., QDL) are precisely the strongly grounded solutions to Quantified Reflective Logic (i.e., QRL). These results apply where variables cross modal scopes in any combination of the following two cases:

(1) where variables are universally quantified precisely over the scope of a default (or equivalently across the scope of all defaults and the initial theory Γ since they are connected by conjunction and since the universal quantifier commutes with conjunction),

(2) where variables are not quantified within the scope of the reflective equivalence in which case they are free within the scope of the theorem schemata proven herein and those schemata lie within the scope of any universal or existential quantification of such variables.

This paper does not address the important case where existential quantification occurs precisely over the scope of one or more defaults nor more complicated systems whereby quantifiers and modal symbols are

(strongly-grounded k) =d $\neg\exists p((k\supseteq p)\wedge(\neg(p\supseteq k))\wedge(p\supseteq(\text{folth}\ p))\wedge(p\supseteq\Gamma)\wedge\wedge_i((\wedge_{j=1,mi}((\neg\beta_{ij})\notin k))\rightarrow(((\alpha_i\varepsilon p)\wedge\wedge_{j=1,mi}((\neg\beta_{ij})\notin p))\rightarrow(\chi_i\varepsilon p))))$. .

[9] [Konolige 1987a 1987b] previously attempted to prove a theorem relating the kernels of the "strongly grounded" fixed-points of Autoepistemic logic to the fixed-points of Default Logic (i.e. dl). A correct version of that attempt is described in [Antoniou 1997]. Since the fixed-points of Reflective Logic (i.e. rl) are the kernels of the fixed-points of Autoepistemic Logic that result is related to the result given herein. However, that result is not as general as the result given herein because it does not explain the relationship between Quantified Default Logic (i.e. QDL) and Quantified Reflective Logic (i.e. QRL) where variables may occur free in the 'α, 'β_{ij}, and 'χ_i sentences or in the sentences in Γ thereby being quantified across the modal scopes of the defaults (which is the subject of this paper).

nested in complex ways. (It is noted, however, that [Brown 1978] showed how an additional modal axiom allows modal scopes to be reduced to a depth of one even in the presence of quantifiers.)

This paper has not addressed automatic deduction systems for QDL and QRL, but there is the obvious point that theorems RD11 and RD12 suggest that a good deduction system for one logic may form the basis for a deduction system for the other logic. In particular, a deduction system that produced the QRL solutions could be used to produce the QDL solutions by checking which of those solutions satisfied a supporting condition (e.g. being strongly grounded) in RD11. The cost of checking a solution once it is produced would seem to be less than the cost of mechanically computing it.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant Numbers: MIP-9526532, EIA-9818341, and EIA-9972843.

Bibliography

[Antoniou 1997] Antoniou, Grigoris 1997. *NonMonotonic Reasoning*, MIT Press.

[Bressan 1972] Bressan, Aldo 1972. *A General Interpreted Modal Calculus,* Yale University Press.

[Brown 1978] Brown,Frank M. 1978. "A Sequent Calculus for Modal Quantificational Logic", *3rd AISB/GI Conference Proceedings,* Hamburg, July 1978.

[Brown 1979] Brown, Frank M. 1979. "A Theorem Prover for Meta theory", *Proceedings Fourth Workshop on Automated Deduction*, Austin, Texas.

[Brown 1987] Brown, Frank M. 1987. "The Modal Logic Z", In *The Frame Problem in AI"; Proc. of the 1987 AAAI Workshop*, Morgan Kaufmann, Los Altos, CA .

[Brown 1989] Brown, Frank M. 1989. "The Modal Quantificational Logic Z Applied to the Frame Problem", advanced paper *First International Workshop on Human & Machine Cognition*, May 1989 Pensacola, Florida. Abbreviated version published in *International Journal of Expert Systems Research and Applications, Special Issue: The Frame Problem. Part A.* eds. Keneth Ford and Pattrick Hayes, vol. 3 number 3, pp169-206 JAI Press 1990. Reprinted in *Reasoning Agents in a Dynamic World: The Frame problem*, editors: Kenneth M. Ford, Patrick J. Hayes, JAI Press 1991.

[Brown 2003a] Frank M, Brown, “Representing Reflective Logic in Modal Logic”, Proceedings of the X-th International Conference "Knowledge – Dialogue – Solution". KDS 2003, June 16-26 2003, Varna, Bulgaria, Sofia FOI-COMMERCE – 2003, ISBN954-16-0025-5.

[Brown 2003b] Frank M, Brown, “Representing Default Logic in Modal Logic”, Proceedings of the X-th International Conference "Knowledge – Dialogue – Solution". KDS 2003, June 16-26 2003, Varna, Bulgaria, Sofia FOI-COMMERCE – 2003, ISBN954-16-0025-5.

[Carnap 1946] Carnap, Rudolf 1946. "Modalities and Quantification" *Journal of Symbolic Logic*, vol. 11, number 2.

[Carnap 1956] Carnap, Rudolf 1956. *Meaning and Necessity: A Study in the Semantics of Modal Logic*, The University of Chicago Press.

[Cocchiarella 1984] Cocchiarella, N. B. 1984. "Philosophical Perspectives on Quantification in Tense and Modal Logic", *Handbook of Philosophical Logic*, D. Reidel.

[Fine 1970] Fine, K. 1970. "Propositional Quantifiers in Modal Logic" *Theoria* 36, p336--346.

[Hendry & Pokriefka 1985] Hendry, Herbert E. & Pokriefka, M. L. 1985. "Carnapian Extensions of S5", *Journal of Phil. Logic 14*.

[Hughes & Cresswell 1968] Hughes, G. E. & Creswell, M. J., 1968. *An Introduction to Modal Logic*, Methuen & Co. Ltd., London.

[Konolige 1987a] Konolige, Kurt 1987. "On the Relation Between Default Theories and Autoepistemic Logic", *IJCAI87 Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, 1987.

[Konolige 1987b] Konolige, Kurt 1987b. "On the Relation Between Default Theories and Autoepistemic Logic", personally circulated new version of IJCAI87 paper correcting it to account for a counterexample found by Gelfond and Przymusinska.

[Lewis 1936] Lewis, C. I. 1936. Strict Implication, *Journal of Symbolic Logic*, vol I.

[Mendelson 1964] Mendelson, E. 1964. *Introduction to Mathematical Logic*, Van Norstrand, Reinhold Co., New York.

[Parks 1976] Parks, Z. 1976. "An Investigation into Quantified Modal Logic", Studia Logica 35, p109-125.

[Reiter 1980] Reiter, R. 1980. "A Logic for Default Reasoning" *Artificial Intelligence*, 13.

Author information

Frank M. Brown- Artificial Intelligence Laboratory, University of Kansas, Lawrence, Kansas, 66045, e-mail: brown@ku.edu.

REPRESENTING AUTOEPISTEMIC LOGIC IN MODAL LOGIC

Frank M. Brown

Abstract: The nonmonotonic logic called Autoepistemic Logic is shown to be representable in a monotonic Modal Quantificational Logic whose modal laws are stronger than S5. Specifically, it is proven that a set of sentences of First Order Logic is a fixed-point of the fixed-point equation of Autoepistemic Logic with an initial set of axioms if and only if the meaning or rather disquotation of that set of sentences is logically equivalent to a particular modal functor of the meaning of that initial set of sentences. This result is important because the modal representation allows the use of powerful automatic deduction systems for Modal Logic and unlike the original Autoepistemic Logic, it is easily generalized to the case where quantified variables may be shared across the scope of modal expressions thus allowing the derivation of quantified consequences. Furthermore, this generalization properly treats such quantifiers since both the Barcan formula and its converse hold

Keywords: Autoepistemic Logic, Modal Logic, Nonmonotonic Logic.

1. Introduction

One of the most well known nonmonotonic logics [Antoniou 1997] which inherently deals with entailment conditions in addition to possibility conditions in its sentences is the so-called Autoepistemic Logic [Moore 1985][10]. The basic idea of Autoepistemic Logic is that there is a set of axioms $\{'\Gamma_i\}$ and for every closed sentence χ there are two non-logical "inference rules" of the forms:

$$\frac{\chi :}{L'\chi} \qquad\qquad \frac{:\neg\chi}{\neg L'\chi}$$

where the predicate symbol L intuitively means that its argument names a sentence which is inferable. The first rule suggests that L $'\chi$ may be inferred from χ and the second rule suggests that $\neg L'\chi$ may be inferred if χ is not inferable. When L is in Γ such "inference rules" maybe circular in that determining if they are applicable depends on the inferability or noninferability of χ which in turn depends on what else was derivable. Thus, tentatively applying such inference rules by checking whether χ has been or has not yet been inferred produces consequences which may later have to be retracted. For this reason valid inferences in a nonmonotonic logic such as Autoepistemic Logic are essentially carried out not in the original nonmonotonic language, but rather in some (monotonic) metatheory in which that nonmonotonic logic is defined. [Moore 1985; Konolige 1987; Konolige 1987b] explicated the above intuition by defining Autoepistemic Logic in terms of the set theoretic proof theory metalanguage of a First Order Logic (i.e. FOL) object language with the fixed-point equation:

$$'\kappa\text{=(ael }'\kappa\ \{'\Gamma_i\}\})$$

where ael is defined as:

$$\text{(ael }'\kappa\ \{'\Gamma_i\})\text{=df(fol(}\{'\Gamma_i\}\cup\{'(\text{L }'\chi_i):'\chi_i\varepsilon'\kappa\}\cup\{'(\neg(\text{L }'\chi_i)):'\chi_i\not\in'\kappa\}))$$

where $'\chi_i$ is the ith sentence of the FOL object language and where $'\kappa$ and $\{'\Gamma_i\}$ are sets of closed sentences of the FOL object language. A closed sentence is a sentence without any free variables. fol is a function which produces the set of theorems derivable in FOL from the set of sentences to which it is applied. The quotations appended to the front of these Greek letters indicate references in the metalanguage to the sentences of the FOL object language. Interpreted doxastically this fixed-point equation states:

the set of closed sentences which are believed is equal to
the set of theorems derivable by the laws of FOL from the union of
the set of closed sentences $\{'\Gamma_i\}$,
the set of all closed sentences of the form: $'(\text{L }'\chi_i)$ for each i such that $'\chi_i$ is believed,
and the set of all closed sentences of the form: $'(\neg(\text{L }'\chi_i))$ for each i such that $'\chi_i$ is not believed.

The purpose of this paper is to show that all this metatheoretic machinery including the formalized syntax of FOL, the proof theory of FOL, the axioms of a strong set theory, and the set theoretic fixed-point equation is

[10] Autoepistemic Logic may be viewed as an improved version of the systems described in [McDermott 1980; McDermott 1982].

not needed and that the essence of Autoepistemic Logic is representable as a necessary equivalence in a (monotonic) Modal Quantificational Logic. Interpreted as a doxastic logic this equivalence states:

that which is believed is equivalent to: for all i Γ_i and for all i (L 'χ_i) if and only if χ_i is believed.

thereby eliminating the metatheoretic machinery.[11]

The remainder of this paper proves that this modal representation is equivalent to Autoepistemic Logic. Section 2 describes a formalized syntax for a FOL object language. Section 3 describes the part of the proof theory of FOL needed herein (i.e. theorems FOL1-FOL4). Section 4 describes the Intensional Semantics of FOL which includes laws giving the meaning of FOL sentences: M0-M8, theorems giving the meaning of sets of sentences: MS1, MS2, MS3, and laws specifying the relationship of meaning and modality to the proof theory of FOL (i.e. the laws R0, A1, A2, and A3 and the theorems: C1, C2, C3, and C4). The modal version of Autoepistemic Logic is defined in section 5 and explicated with theorems MA1-MA6 and SS1-SS2. In section 6, this modal version is shown by theorems AEL1 and AEL2 to be equivalent to the set theoretic fixed-point equation for Autoepistemic Logic. Figure 1 outlines the relationship of all these theorems in producing the final theorems AEL2, FOL4, and MA6.

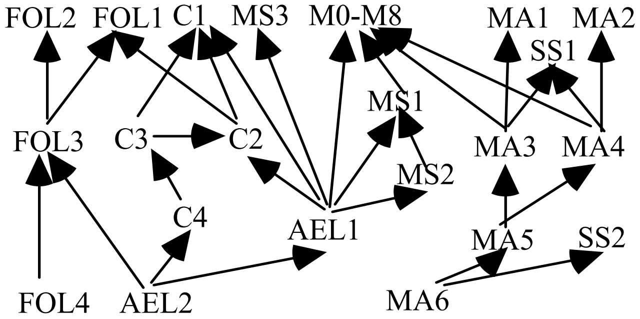


Figure 1: Dependencies among the Theorems

2. Formal Syntax of First Order Logic

We use a First Order Logic (i.e. FOL) defined as the six tuple: ($\rightarrow$, #f, $\forall$, *vars*, *predicates*, *functions*) where $\rightarrow$, #f, and $\forall$ are logical symbols, *vars* is a set of variable symbols, *predicates* is a set of predicate symbols each of which has an implicit arity specifying the number of associated terms, and *functions* is a set of function symbols each of which has an implicit arity specifying the number of associated terms. The sets of logical symbols, variables, predicate symbols, and function symbols are pairwise disjoint. Lower case Roman letters possibly indexed with digits are used as variables. Greek letters possibly indexed with digits are used as syntactic metavariables. γ, $\gamma_1,...,\gamma_n$, range over the variables, ξ, $\xi_1,...,\xi_n$ range over sequences of variables of an appropriate arity, $\pi,\pi_1...\pi_n$ range over the predicate symbols, ϕ, $\phi_1...\phi_n$ range over function symbols, δ, $\delta_1...\delta_n$, σ range over terms, and α, $\alpha_1,...,\alpha_n$, β, $\beta_1,...,\beta_n,\chi$, $\chi_1,...,\chi_n$, $\Gamma_1,...,\Gamma_n,\varphi$ range over sentences. The terms are of the forms γ and (ϕ δ1...δn), and the sentences are of the forms ($\alpha\rightarrow\beta$), #f, ($\forall\gamma$ α), and (π $\delta_1...\delta_n$). A nullary predicate π or function ϕ is written as a sentence or a term without parentheses. $\varphi\{\pi/\lambda\xi\alpha\}$ represents the replacement of all occurrences of π in φ by $\lambda\xi\alpha$ followed by lambda conversion. The primitive symbols are shown in Figure 2 with their intuitive interpretations.

Symbol	Meaning
$\alpha\rightarrow\beta$	if α then β.
#f	falsity
$\forall\gamma$ α	for all γ, α.

Figure 2: Primitive Symbols of First Order Logic

The defined symbols are listed in Figure 3 with their definitions and intuitive interpretations.

[11] The occurrence of quotation in the argument to L may be replaced by using a new symbol *L* such that (*L*) replaces (L ').

Symbol	Definition	Meaning		Symbol	Definition	Meaning
$\neg\alpha$	$\alpha \rightarrow \#f$	not α		$\alpha\wedge\beta$	$\neg(\alpha \rightarrow \neg\beta)$	α and β
#t	$\neg$ #f	truth		$\alpha\leftrightarrow \beta$	$(\alpha\rightarrow \beta) \wedge (\beta\rightarrow \alpha)$	α if and only if β
$\alpha\vee\beta$	$(\neg \alpha)\rightarrow \beta$	α or β		$\exists\gamma\, \alpha$	$\neg\forall\gamma\, \neg\alpha$	for some γ , α

Figure 3: Defined Symbols of First Order Logic

The particular FOL used herein includes the predicate symbol L and a denumerably infinite number of 0-ary function symbols representing the names (i.e. '□) of the sentences (i.e. □) of this First Order Logic. The FOL object language expressions are referred in the metalanguage (which also includes a FOL syntax) by inserting a quote sign in front of the object language entity thereby making a structural descriptive name of that entity. In addition to referring to object language sentences, the formalized metalanguage also needs to refer to sets of sentences of FOL. Generally, a set of sentences is represented as: {'Γ_i} which is defined as: {'Γ_i: #t} which in turn is defined as: {s: $\exists$i(s='Γ_i)} where i ranges over some range of numbers (which may be finite or non-infinite). With a slight abuse of notation we also write 'κ, 'Γ to refer to such sets.

3. Proof Theory of First Order Logic

First Order Logic (i.e. FOL) is axiomatized with a recursively enumerable set of theorems as the set of axioms is itself recursively enumerable and its inference rules are recursive. The axioms and inference rules of FOL [Mendelson 1964] are those given in Figure 4. They form a standard set of axioms and inference rules for FOL.

MA1: $\alpha \rightarrow (\beta\rightarrow \alpha)$ MR1: from α and $(\alpha\rightarrow \beta)$ infer β

MA2: $(\alpha\rightarrow (\beta\rightarrow \rho)) \rightarrow ((\alpha\rightarrow \beta)\rightarrow (\alpha\rightarrow \rho))$ MR2: from α infer $(\forall\gamma\, \alpha)$

MA3: $((\neg \alpha)\rightarrow (\neg \beta))\rightarrow (((\neg \alpha)\rightarrow \beta)\rightarrow\alpha)$

MA4: $(\forall\gamma\, \alpha)\rightarrow \beta$ where β is the result of substituting an expression (which is free for the free positions of γ in α) for all the free occurrences of γ in α.

MA5: $((\forall\gamma(\alpha \rightarrow \beta)) \rightarrow (\alpha\rightarrow(\forall\gamma\, \beta)))$ where γ does not occur in α.

Figure 4: Inferences Rules and Axioms of FOL

In order to talk about sets of sentences we include in the metatheory set theory symbolism as developed along the lines of [Quine 1969]. This set theory includes the symbols ε, $\not\in$, $\supseteq$, $=$, $\cup$ as is defined therein.
The derivation operation (i.e. fol) of any First Order Logic obeys the Inclusion (i.e. FOL1) and Idempotence (i.e. FOL2) properties:

FOL1: (fol 'κ)$\supseteq$'κ Inclusion

FOL2: (fol 'κ)$\supseteq$(fol(fol 'κ)) Idempotence

From these two properties we prove:

FOL3: (ael 'κ 'Γ 'α_i:'β_{ij}/'χ_i))=(fol(ael 'κ 'Γ 'α_i:'β_{ij}/'χ_i))

proof: FOL1 and FOL2 imply that (fol(fol 'κ))=(fol 'κ). Since ael begins with fol this implies: 'κ=(fol(ael 'κ)) QED.

FOL4: ('κ=(ael 'κ 'Γ 'α_i:'β_{ij}/'χ_i))$\rightarrow$('κ=(fol 'κ))

proof: From the hypothesis and FOL3: 'κ=(fol(ael 'κ 'Γ 'α_i:'β_{ij}/'χ_i)) is derived. Using the hypothesis to replace (ael 'κ 'Γ 'α_i:'β_{ij}/'χ_i) by 'κ in this result gives: 'κ=(fol 'κ). QED.

4. Intensional Semantics of FOL

The meaning (i.e. mg) [Brown 1978, Boyer&Moore 1981] or rather disquotation of a sentence of First Order Logic (i.e. FOL) is defined to satisfy the laws given in Figure 5 below mg is defined in terms of mgs which maps each FOL object language sentence and an association list into a meaning. Likewise, mgn maps a FOL object language term and an association list into a meaning. An association list is simply a list of pairs consisting of an object language variable and the meaning to which it is bound.

M0: (mg 'α) =df (mgs '($\forall\gamma 1 \ldots \gamma n\ \alpha$)'()) where '$\gamma$1...'$\gamma$n are all the free variables in 'α

M1: (mgs '($\alpha \to \beta$)a) $\leftrightarrow$ ((mgs 'α a)$\to$(mgs 'β a))

M2: (mgs '#f a) $\leftrightarrow$ #f

M3: (mgs '($\forall\ \gamma\ \alpha$)a) $\leftrightarrow$ $\forall$x(mgs 'α(cons(cons 'γ x)a))

M4: (mgs '($\pi\ \delta_1 \ldots \delta_n$)a) $\leftrightarrow$ (π(mgn 'δ_1 a)...(mgn 'δ_n a)) for each predicate symbol 'π.

M5: (mgn '($\phi\ \ \delta_1 \ldots \delta_n$)a) = ($\phi$(mgn '$\delta_1$ a)...(mgn 'δ_n a)) for each function symbol 'ϕ.

M6: (mgn 'γ a) = (cdr(assoc 'γ a))

M7: (assoc v L) = (if(eq? v(car(car L))) (car L) (assoc v(cdr L)))
where: cons, car, cdr, eq?, if are axiomatized as they are axiomatized in Scheme.

M8: (mgn ''α a) = 'α

Figure 5: The Meaning of FOL Sentences

The meaning of a set of sentences is defined in terms of the meanings of the sentences in the set as:

(ms 'κ) =df $\forall$s((sε'κ)$\to$(mg s))

MS1: (ms{'α: Γ}) $\leftrightarrow$ $\forall\xi(\Gamma\to\alpha)$ where ξ is the sequence of all the free variables in 'α and where Γ is any sentence of the intensional semantics.

proof: (ms{'α:Γ}) Unfolding ms and the set pattern abstraction symbol gives: $\forall$s((sε{s: $\exists\xi$((s='α)$\wedge\Gamma$)})$\to$(mg s))

where ξ is a sequence of the free variables in 'a. This is equivalent to: $\forall$s(($\exists\xi$((s='α)$\wedge\Gamma$)))$\to$(mg s))

which is logically equivalent to: $\forall$s$\forall\xi$ (((s='κ)$\wedge\Gamma$)$\to$(mg s)) which is equivalent to: $\forall\xi$($\Gamma\to$(mg 'α))

Unfolding mg using M0-M8 then gives: $\forall\xi(\Gamma\to\alpha)$ QED

The meaning of the union of two sets of FOL sentences is the conjunction of their meanings (i.e. MS3) and the meaning of a set is the meaning of all the sentences in the set (i.e. MS2):

MS2: (ms{'Γ_i}) $\leftrightarrow$ $\forall i\forall\xi_i\Gamma_i$

proof: (ms{'Γ_i}) Unfolding the set notation gives: (ms{'Γ_i: #t})

By MS1 this is equivalent to: $\forall_i\forall\xi_i$(#t$\to\Gamma_i$) which is equivalent to: $\forall i\forall\xi_i(\Gamma_i$ QED.

MS3: (ms('$\kappa\cup$'Γ)) $\leftrightarrow$ ((ms 'κ)$\wedge$(ms 'Γ))

proof: Unfolding ms and union in: (ms('$\kappa\cup$'Γ)) gives: $\forall$s((sε{s: (sε'κ)$\vee$(sε'Γ)})$\to$(mg s)) or rather:

$\forall$s(((sε'κ)$\vee$(sε'Γ))$\to$(mg s)) which is logically equivalent to: ($\forall\alpha$((sε'κ)$\to$(mg s)))$\wedge$($\forall$s((sε'Γ)$\to$(mg s)))

Folding ms twice then gives:((ms 'κ)$\wedge$(ms 'Γ)) QED.

The meaning operation may be used to develop an Intensional Semantics for a FOL object language by axiomatizing the modal concept of necessity so that it satisfies the theorem:

C1: ('$\alpha\varepsilon$(fol 'κ)) $\leftrightarrow$ ([] ((ms 'κ)$\to$(mg 'α)))

for every sentence 'α and every set of sentences 'κ of that FOL object language. The necessity symbol is represented by a box: []. C1 states that a sentence of FOL is a FOL-theorem (i.e. fol) of a set of sentences of FOL if and only if the meaning of that set of sentences necessarily implies the meaning of that sentence. One modal logic which satisfies C1 for FOL is the Z Modal Quantificational Logic described in [Brown 1987; Brown 1989] whose theorems are recursively enumerable and which extends the weaker possibility axioms used in [Lewis 1936; Bressan 1972; Hendry & Pokriefka 1985]. We note that Z includes all the laws of S5 modal Logic [Hughes & Cresswell 1968] whose modal axioms and inference rules are given in Figure 6. Therein, κ and Γ represent arbitrary sentences of the intentional semantics.

R0: from α infer ([] κ)

A2: ([]($\kappa\to\Gamma$)) $\to$ (([]κ)$\to$ ([]Γ))

A1: ([]κ) $\to$ κ

A3: ([]κ) $\vee$ ([]$\neg$[]κ)

Figure 6: The Laws of S5 Modal Logic

These S5 modal laws and the laws of FOL given in Figure 4 constitute an S5 Modal Quantificational Logic similar to [Carnap 1946; Carnap 1956], and a FOL version [Parks 1976] of [Bressan 1972] in which the Barcan formula: ($\forall\gamma$([]κ))$\to$([]$\forall\gamma\kappa$) and its converse hold. The R0 inference rule implies that anything derivable in the metatheory is necessary. Thus, in any logic with R0, contingent facts would never be asserted as additional axioms of the metatheory. For example, we would not assert $([](\kappa\leftrightarrow\Gamma))$ as an axiom and then try to prove $([](\kappa\rightarrow\alpha))$. Instead we would try to prove that $([](\kappa\leftrightarrow\Gamma))\rightarrow([](\kappa\rightarrow\alpha))$.

The defined Modal symbols used herein are listed in Figure 7 with their definitions and interpretations.

Symbol	Definition	Meaning		Symbol	Definition	Meaning
$<>\kappa$	$\neg\ []\ \neg\kappa$	α is logically possible		$[\kappa]\ \Gamma$	$[]\ (\kappa\rightarrow\Gamma)$	β entails α
$\kappa\equiv\Gamma$	$[]\ (\kappa\leftrightarrow\Gamma)$	α is logically equivalent to β		$<\kappa>\ \Gamma$	$<>\ (\kappa\wedge\Gamma)$	α and β is logically possible

Figure 7: Defined Symbols of Modal Logic

For example, folding the definition of entailment, C1 may be rewritten more compactly as:

C1': $('\alpha\varepsilon(\text{fol }'\kappa)) \leftrightarrow ([(\text{ms }'\kappa)](\text{mg }'\alpha))$

This compact notation for entailment is used hereafter.

From the laws of the Intensional Semantics we prove that the meaning of the set of FOL consequences of a set of sentences is the meaning of that set of sentences (C2), the FOL consequences of a set of sentences contain the FOL consequences of another set if and only if the meaning of the first set entails the meaning of the second set (C3), and the sets of FOL consequences of two sets of sentences are equal if and only if the meanings of the two sets are logically equivalent (C4):

C2: $(\text{ms}(\text{fol }'\kappa))\equiv(\text{ms }'\kappa)$

proof: The proof divides into two cases:

(1) $[(\text{ms }'\kappa)](\text{ms}(\text{fol }'\kappa)))$ Unfolding the second ms gives: $[(\text{ms }'\kappa)]\forall s((s\varepsilon(\text{fol }'\kappa))\rightarrow(\text{mg } s))$

By the soundness part of C1 this is equivalent to: $[(\text{ms }'\kappa)]\forall s(([(\text{ms }'\kappa)](\text{mg } s))\rightarrow(\text{mg } s))$

By the S5 laws this is equivalent to: $\forall s(([(\text{ms }'\kappa)](\text{mg } s))\rightarrow [(\text{ms }'\kappa)](\text{mg } s))$ which is a tautology.

(2) $[(\text{ms}(\text{fol }'\kappa))](\text{ms }'\kappa)$ Unfolding ms twice gives: $[\forall s((s\varepsilon(\text{fol }'\kappa))\rightarrow(\text{mg } s))]\forall s((s\varepsilon'\kappa)\rightarrow(\text{mg } s))$

which is: $[\forall s((s\varepsilon(\text{fol }'\kappa))\rightarrow(\text{mg } s))]((s\varepsilon'\kappa)\rightarrow(\text{mg } s))$ Backchaining on the hypothesis and then dropping it gives: $(s\varepsilon'\kappa)\rightarrow(s\varepsilon(\text{fol }'\kappa))$. Folding $\supseteq$ gives an instance of FOL1. QED.

C3: $(\text{fol }'\kappa)\supseteq(\text{fol }'\Gamma) \leftrightarrow ([(\text{ms }'\kappa)](\text{ms }'\Gamma))$

proof: Unfolding $\supseteq$ gives: $\forall s((s\varepsilon(\text{fol }'\Gamma))\rightarrow(s\varepsilon(\text{fol }'\kappa)))$

By C1 twice this is equivalent to: $\forall s(([(\text{ms }'\Gamma)](\text{mg } s))\rightarrow([(\text{ms }'\kappa)](\text{mg } s)))$

By the laws of S5 modal logic this is equivalent to: $([(\text{ms }'\kappa)]\forall s(([(\text{ms }'\Gamma)](\text{mg } s)))\rightarrow(\text{mg } s)))$

By C1 this is equivalent to: $[(\text{ms }'\kappa)]\forall s((s\varepsilon(\text{fol }'\Gamma))\rightarrow(\text{mg } s))$. Folding ms then gives: $[(\text{ms }'\kappa)](\text{ms}(\text{fol }'\Gamma))$

By C2 this is equivalent to: $[(\text{ms }'\kappa)](\text{ms }'\Gamma)$. QED.

C4: $((\text{fol }'\kappa)=(\text{fol }'\Gamma)) \leftrightarrow ((\text{ms }'\kappa)\equiv(\text{ms }'\Gamma))$

proof: This is equivalent to $(((\text{fol }'\kappa)\supseteq(\text{fol }'\Gamma))\wedge((\text{fol }'\Gamma)\supseteq(\text{fol }'\kappa))) \leftrightarrow ([(\text{ms }'\kappa)](\text{ms }'\Gamma))\wedge([(\text{ms }'\Gamma)](\text{ms }'\kappa))$ which follows by using C3 twice.

5. Autoepistemic Logic Represented in Modal Logic

The fixed-point equation for Autoepistemic Logic may be expressed in S5 Modal Quantificational Logic by the necessary equivalence:

$$\kappa\equiv(\text{AEL }\kappa\ \Gamma)$$

where AEL is defined as follows: $(\text{AEL }\kappa\ \Gamma)$ =df $\Gamma\wedge\forall i((L\ '\chi_i)\leftrightarrow([\kappa]\chi_i))$

where χ_i is the ith sentence of the FOL object language.

Given below are some simple properties of AEL used to prove the equivalence of the proof theoretic and modal representations of Autoepistemic Logic. The first two theorems state that AEL entails Γ and that AEL entails for all i, $(L\ '\chi_i)$ if and only if χ_i holds in κ.

MA1: $[(\mathrm{AEL}\ \kappa\ \Gamma)]\Gamma$

proof: By R0 it suffices to prove: $(\mathrm{AEL}\ \kappa\ \Gamma)\rightarrow\Gamma$. Unfolding AEL gives: $(\Gamma\wedge\forall i((\mathrm{L}\ '\chi_i)\leftrightarrow([\kappa]\chi_i)))\rightarrow\Gamma$ which is a tautology. QED.

MA2: $[(\mathrm{AEL}\ \kappa\ \Gamma)]\forall i((\mathrm{L}\ '\chi_i)\leftrightarrow([\kappa]\chi_i))$

proof: By R0 it suffices to prove: $(\mathrm{AEL}\ \kappa\ \Gamma)\rightarrow\forall i((\mathrm{L}\ '\chi_i)\leftrightarrow([\kappa]\chi_i))$

Unfolding AEL gives: $[\Gamma\wedge\forall i((\mathrm{L}\ '\chi_i)\leftrightarrow([\kappa]\chi_i)))\rightarrow\forall i((\mathrm{L}\ '\chi_i)\leftrightarrow([\kappa]\chi_i)))$ which is a tautology. QED.

The concept (i.e. ss) of the combined meaning of all the sentences of the FOL object language whose meanings are entailed by a proposition is defined as follows: $(\mathrm{ss}\ \kappa)$ =df $\forall s(([\kappa](\mathrm{mg}\ s))\rightarrow(\mathrm{mg}\ s))$. SS1 shows that a proposition entails the combined meaning of the FOL object language sentences that it entails. SS2 shows that if a proposition is necessarily equivalent to the combined meaning of all the FOL object language sentences that it entails, then there exists a set of FOL object language sentences whose meaning is necessarily equivalent to that proposition:

SS1: $[\kappa](\mathrm{ss}\ \kappa)$

proof: By R0 it suffices to prove: $\kappa\rightarrow(\mathrm{ss}\ \kappa)$. Unfolding ss gives: $\kappa\rightarrow\forall s(([\kappa](\mathrm{mg}\ s))\rightarrow(\mathrm{mg}\ s))$

which is equivalent to:$\forall s(([\kappa](\mathrm{mg}\ s))\rightarrow(\kappa\rightarrow(\mathrm{mg}\ s)))$ which is an instance of A1. QED.

SS2: $(\kappa\equiv(\mathrm{ss}\ \kappa))\rightarrow\ \exists s(\kappa\equiv(\mathrm{ms}\ s))$

proof: Letting s be {s: $([\kappa](\mathrm{mg}\ s))$ gives: $(\kappa\equiv(\mathrm{ss}\ \kappa))\rightarrow\ (\kappa\equiv(\mathrm{ms}\{s:\ ([\kappa](\mathrm{mg}\ s))))$

Unfolding ms and lambda conversion gives: $(\kappa\equiv(\mathrm{ss}\ \kappa))\leftrightarrow\ (\kappa\equiv\forall s(([\kappa](\mathrm{mg}\ s))\rightarrow(\mathrm{mg}\ s)))$

Folding ss gives a tautology. QED.

Theorems MA3 and MA4 are analogous to MA1 and MA2 except that AEL is replaced by the combined meaning of all of the sentences entailed by AEL.

MA3: $[\mathrm{ss}(\mathrm{AEL}\ \kappa\ \forall i\Gamma_i)]\forall i\Gamma_i$

proof: By R0 it suffices to prove: $(\mathrm{ss}(\mathrm{AEL}\ \kappa\ \forall i\Gamma_i))\rightarrow\forall i\Gamma_i$

Unfolding ss gives: $(\forall s(([(\mathrm{AEL}\ \kappa\ \forall i\Gamma_i)](\mathrm{mg}\ s))\rightarrow(\mathrm{mg}\ s)))\rightarrow\forall i\Gamma_i$

which is equivalent to: $(\forall s(([(\mathrm{AEL}\ \kappa\ \forall i\Gamma_i)](\mathrm{mg}\ s))\rightarrow(\mathrm{mg}\ s)))\rightarrow\Gamma_i$

which by the meaning laws is equivalent to: $(\forall s(([(\mathrm{AEL}\ \kappa\ \forall i\Gamma_i)](\mathrm{mg}\ s))\rightarrow(\mathrm{mg}\ s)))\rightarrow(\mathrm{mg}\ '\Gamma_i)$

Backchaining on $(\mathrm{mg}\ \ '\Gamma_i)$ with s in the hypothesis assigned to be $'\Gamma_i$ in the conclusion shows that it suffices to prove: $([(\mathrm{AEL}\ \kappa\ \forall i\Gamma_i)](\mathrm{mg}\ '\Gamma_i))$ which by the meaning laws is equivalent to: $([(\mathrm{AEL}\ \kappa\ \forall i\Gamma_i)]\Gamma_i)$

which by the laws of S5 Modal Logic is equivalent to: $([(\mathrm{AEL}\ \kappa\ \forall i\Gamma_i)]\forall i\Gamma_i)$ which is an instance of MA1. QED.

MA4: $[(\mathrm{ss}(\mathrm{AEL}\ \kappa\ \Gamma))]\forall i((\mathrm{L}\ '\chi_i)\leftrightarrow([\kappa]\chi_i))$

proof: By R0 it suffices to prove: $(\mathrm{ss}(\mathrm{AEL}\ \kappa\ \Gamma))\rightarrow\forall i((\mathrm{L}\ '\chi_i)\leftrightarrow([\kappa]\chi_i))$

which is equivalent to: $(\mathrm{ss}(\mathrm{AEL}\ \kappa\ \Gamma))\rightarrow((([\kappa]\chi_i)\rightarrow(\mathrm{L}\ '\chi_i))\wedge((\neg([\kappa]\chi_i))\rightarrow(\neg(\mathrm{L}\ '\chi_i))))$

Unfolding ss gives: $(\forall s(([(\mathrm{AEL}\ \kappa\ \Gamma)](\mathrm{mg}\ s))\rightarrow(\mathrm{mg}\ s)))\rightarrow((([\kappa]\chi_i)\rightarrow(\mathrm{L}\ '\chi_i))\wedge((\neg([\kappa]\chi_i))\rightarrow(\neg(\mathrm{L}\ '\chi_i))))$

Letting the quantified s in the hypothesis have the two instances: $'(\mathrm{L}\ '\chi_i)$ and $'(\neg(\mathrm{L}\ '\chi_i))$ and then dropping that hypothesis gives:

$((([(\mathrm{AEL}\ \kappa\ \Gamma)](\mathrm{mg}\ '(\mathrm{L}\ '\chi_i)))\rightarrow(\mathrm{mg}\ '(\mathrm{L}\ '\chi_i)))\wedge(([(\mathrm{AEL}\ \kappa\ \Gamma)](\mathrm{mg}\ '(\neg(\mathrm{L}\ '\chi_i))))\rightarrow(\mathrm{mg}\ '(\neg(\mathrm{L}\ '\chi_i)))))$

$\rightarrow((([\kappa]\chi_i)\rightarrow(\mathrm{L}\ '\chi_i))\wedge((\neg([\kappa]\chi_i))\rightarrow(\neg(\mathrm{L}\ '\chi_i))))$

By the meaning laws M0-M8 this is equivalent to:

$((([(\mathrm{AEL}\ \kappa\ \Gamma)](\mathrm{L}\ '\chi_i))\rightarrow(\mathrm{L}\ '\chi_i))\wedge(([(\mathrm{AEL}\ \kappa\ \Gamma)](\neg(\mathrm{L}\ '\chi_i)))\rightarrow(\neg(\mathrm{L}\ '\chi_i))))\rightarrow((([\kappa]\chi_i)\rightarrow(\mathrm{L}\ '\chi_i))\wedge((\neg([\kappa]\chi_i))\rightarrow(\neg(\mathrm{L}\ '\chi_i))))$

Using these instances of the hypothesis to backchain on $(\mathrm{L}\ '\chi_i)$ and $(\neg(\mathrm{L}\ '\chi_i))$ in the conclusion, and then dropping these instances gives:

$((([\kappa]\chi_i)\rightarrow([(\mathrm{AEL}\ \kappa\ \Gamma)](\mathrm{L}\ '\chi_i)))\wedge((\neg([\kappa]\chi_i)))\rightarrow([(\mathrm{AEL}\ \kappa\ \Gamma)](\neg(\mathrm{L}\ '\chi_i))))$

Using the laws of S5 Modal Logic then gives: $([(\mathrm{AEL}\ \kappa\ \Gamma)]((([\kappa]\chi_i)\rightarrow(\mathrm{L}\ '\chi_i))\wedge((\neg([\kappa]\chi_i))\rightarrow(\neg(\mathrm{L}\ '\chi_i))))$

which is equivalent to: $[(\mathrm{AEL}\ \kappa\ \Gamma)]((\mathrm{L}\ '\chi_i)\leftrightarrow([\kappa]\chi_i))$ which holds by MA2. QED.

Finally MA5 and MA6 show that talking about the meanings of sets of FOL sentences in the modal representation of Autoepistemic Logic is equivalent to talking about propositions in general.

MA5: $(\mathrm{ss}(\mathrm{AEL}\ \kappa\ \forall i\Gamma_i)) \leftrightarrow (\mathrm{AEL}\ \kappa\ \forall i\Gamma_i)$

proof: In view of SS1, it suffices to prove: $(\mathrm{ss}(\mathrm{AEL}\ \kappa\ \forall i\Gamma_i))\rightarrow(\mathrm{AEL}\ \kappa\ \forall i\Gamma_i)$

Unfolding the second occurrence of AEL gives: $(\mathrm{ss}(\mathrm{AEL}\ \kappa\ \forall i\Gamma_i))\rightarrow(\forall i\Gamma_i\wedge\forall i((\mathrm{L}\ '\chi_i)\leftrightarrow([\kappa]\chi_i)))$

which holds by theorems MA3 and MA4. QED.

MA6: $(\kappa\equiv(\mathrm{AEL}\ \kappa\ \forall i\Gamma_i)))\rightarrow\exists s(\kappa\equiv(\mathrm{ms}\ s))$

proof: $(\kappa\equiv(\mathrm{ss}(\mathrm{AEL}\ \kappa\ \forall i(\mathrm{mg}\ '\Gamma_i))))$ is derived from the hypothesis and MA5. Using the hypothesis to replace $(\mathrm{AEL}\ \kappa\ \forall i(\mathrm{mg}\ '\Gamma_i)))$ by κ in this result gives: $(\kappa\equiv(\mathrm{ss}\ \kappa))$. By SS2 this implies the conclusion. QED.

6. Conclusion: Autoepistemic Logic represented in Modal Logic

The relationship between the proof theoretic definition of Autoepistemic Logic [Moore 1985] and the modal representation is proven in two steps. First theorem AEL1 shows that the meaning of the set ael is the proposition AEL and then theorem AEL2 shows that a set of FOL sentences which contains its FOL theorems is a fixed-point of the fixed-point equation of Autoepistemic Logic with an initial set of axioms if and only if the meaning (or rather disquotation) of that set of sentences is logically equivalent to AEL of the meanings of that initial set of sentences.

AEL1: $(\mathrm{ms}(\mathrm{ael}(\mathrm{fol}\ '\kappa)\{'\Gamma_i\}))\equiv(\mathrm{AEL}(\mathrm{ms}\ '\kappa)(\forall i\Gamma_i))$

proof: By R0 it suffices to prove: $(\mathrm{ms}(\mathrm{ael}(\mathrm{fol}\ '\kappa)\{'\Gamma_i\}))\leftrightarrow(\mathrm{AEL}(\mathrm{ms}\ '\kappa)\Gamma)$. The left side is: $\mathrm{ms}(\mathrm{ael}(\mathrm{fol}\ '\kappa)\{'\Gamma_i\})$

Unfolding the definition of ael gives: $\mathrm{ms}(\mathrm{fol}(\{'\Gamma_i\}\cup\{'(\mathrm{L}\ '\chi_i)\text{: }'\chi_i\varepsilon(\mathrm{fol}\ '\kappa)\}\cup\{'(\neg(\mathrm{L}\ '\chi_i))\text{: }'\chi_i\notin(\mathrm{fol}\ '\kappa)\}))$

By C2 this is equivalent to: $\mathrm{ms}(\{'\Gamma_i\}\cup\{'(\mathrm{L}\ '\chi_i)\text{: }'\chi_i\varepsilon(\mathrm{fol}\ '\kappa)\}\cup\{'(\neg(\mathrm{L}\ '\chi_i))\text{: }'\chi_i\notin(\mathrm{fol}\ '\kappa)\})$

Using C1 twice gives: $\mathrm{ms}(\{'\Gamma_i\}\cup\{'(\mathrm{L}\ '\chi_i)\text{: }([\mathrm{ms}\ '\kappa]\chi_i)\}\cup\{'(\neg(\mathrm{L}\ '\chi_i))\text{: }\neg([\mathrm{ms}\ '\kappa]\chi_i)\})$

Using MS3 twice gives: $(\mathrm{ms}\{'\Gamma_i\})\wedge(\mathrm{ms}\{'(\mathrm{L}\ '\chi_i)\text{: }([\mathrm{ms}\ '\kappa]\chi_i)\})\wedge(\mathrm{ms}\{'(\neg(\mathrm{L}\ '\chi_i))\text{: }\neg([\mathrm{ms}\ '\kappa]\chi_i)\})$

Using MS2 gives: $(\forall i\Gamma_i)\wedge(\mathrm{ms}\{'(\mathrm{L}\ '\chi_i)\text{: }([\mathrm{ms}\ '\kappa]\chi_i)\})\wedge(\mathrm{ms}\{'(\neg(\mathrm{L}\ '\chi_i))\text{: }\neg([\mathrm{ms}\ '\kappa]\chi_i)\})$

Applying MS1 twice gives: $(\forall i\Gamma_i)\wedge\forall i(([\mathrm{ms}\ '\kappa]\chi_i)\rightarrow(\mathrm{L}\ '\chi_i))\wedge\forall i((\neg([\mathrm{ms}\ '\kappa]\chi_i))\rightarrow(\neg(\mathrm{L}\ '\chi_i)))$

which is logically equivalent to: $(\forall i\Gamma_i)\wedge\forall i((\mathrm{L}\ '\chi_i)\leftrightarrow([\mathrm{ms}\ '\kappa]\chi_i))$

Folding the definition of AEL gives: $(\mathrm{AEL}(\mathrm{ms}\ '\kappa)(\forall i\Gamma_i))$ QED.

AEL2: $((\mathrm{fol}\ '\kappa)=(\mathrm{ael}(\mathrm{fol}\ '\kappa)\{'\Gamma_i\}))\leftrightarrow((\mathrm{ms}\ '\kappa)\equiv(\mathrm{AEL}(\mathrm{ms}\ '\kappa)(\forall i\Gamma_i)))$

proof: $(\mathrm{fol}\ '\kappa)=(\mathrm{ael}(\mathrm{fol}\ '\kappa)\{'\Gamma_i\})$. By FOL3 this is equivalent to: $(\mathrm{fol}\ '\kappa)=(\mathrm{fol}(\mathrm{ael}(\mathrm{fol}\ '\kappa)\{'\Gamma_i\}))$

By C4 this is equivalent to: $(\mathrm{ms}\ '\kappa)\equiv(\mathrm{ms}(\mathrm{ael}(\mathrm{fol}\ '\kappa)\{'\Gamma_i\}))$.

By AEL1 this is equivalent to:$(\mathrm{ms}\ '\kappa)\equiv(\mathrm{AEL}(\mathrm{ms}\ '\kappa)(\forall i\Gamma_i))$ QED.

Theorem AEL2 shows that the set of theorems: $(\mathrm{fol}\ '\kappa)$ of a set $'\kappa$ is a fixed-point of Autoepistemic Logic if and only if the meaning $(\mathrm{ms}\ '\kappa)$ of $'\kappa$ is a solution to the necessary equivalence. Furthermore, by FOL4 there are no other fixed-points (such as a set not containing all its theorems) and by MA6 there are no other solutions (such as a proposition not representable as a sentence in the First Order Logic object language). Therefore the Modal representation of Autoepistemic Logic (i.e. AEL), faithfully represents the original set theoretic description of Autoepistemic Logic (i.e. ael). Finally, we note that $(\mathrm{ms}\ '\kappa)$ and $\forall i\Gamma_i$ may be generalized to be arbitrary propositions κ and Γ giving the more general modal representation: $\kappa\equiv(\mathrm{AEL}\ \kappa\ \Gamma)$.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant Numbers: MIP-9526532, EIA-9818341, and EIA-9972843.

Bibliography

[Antoniou 1997] Antoniou, Grigoris 1997. *NonMonotonic Reasoning*, MIT Press.

[Boyer&Moore 1981] R. S. Boyer and J. Strother Moore, "Metafunctions: proving them correct and using them efficiently as new proof procedures," *The Correctness Problem in Computer Science,* R. S. Boyer and J. Strother Moore, eds., Academic Press, New York, 1981.

[Bressan 1972] Bressan, Aldo 1972. *A General Interpreted Modal Calculus,* Yale University Press.

[Brown 1978] Brown,F.M., "A Semantic Theory for Logic Programming", *Colloquia Mathematica Societatis Janos Bolyai 26*, *Mathematical Logic in Computer Science*, Salgotarjan, Hungry, 1978.

[Brown 1987] Brown, Frank M. 1987. "The Modal Logic Z", In *The Frame Problem in AI*"*; Proc. of the 1987 AAAI Workshop*, Morgan Kaufmann, Los Altos, CA .

[Brown 1989] Brown, Frank M. 1989. "The Modal Quantificational Logic Z Applied to the Frame Problem", advanced paper *First International Workshop on Human & Machine Cognition*, May 1989 Pensacola, Florida. Abreviated version published in *International Journal of Expert Systems Research and Applications, Special Issue: The Frame Problem. Part A.* eds. Keneth Ford and Pattrick Hayes, vol. 3 number 3, pp169-206 JAI Press 1990. Reprinted in *Reasoning Agents in a Dynamic World: The Frame problem*, editors: Kenneth M. Ford, Patrick J. Hayes, JAI Press 1991.

[Carnap 1946] Carnap, Rudolf 1946. "Modalities and Quantification" *Journal of Symbolic Logic*, vol. 11, number 2, 1946.

[Carnap 1956] Carnap, Rudolf 1956. *Meaning and Necessity: A Study in the Semantics of Modal Logic*, The University of Chicago Press.

[Hendry & Pokriefka 1985] Hendry, Herbert E. and Pokriefka, M. L. 1985. "Carnapian Extensions of S5", *Journal of Phil. Logic 14*.

[Hughes & Cresswell 1968] Hughes, G. E. and Creswell, M. J., 1968. *An Introduction to Modal Logic*, Methuen & Co. Ltd., London

[Konolige 1987] Konolige, Kurt 1987. "On the Relation Between Default Theories and Autoepistemic Logic", *IJCAI87 Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, 1987.

[Konolige 1987b] Konolige, Kurt 1987. "On the Relation Between Default Logic and Autoepistemic Theories", Artificial Intelligence 35(3):343-382.

[Lewis 1936] Lewis, C. I. 1936. Strict Implication, *Journal of Symbolic Logic*, vol I.

[McDermott 1980] McDermott, D. and Doyle, J. 1980. "Nonmonotonic Logic I" *Artificial Intelligence*, 13.

[McDermott 1982] McDermott, D. 1982. "Nonmonotonic Logic II: Nonmonotonic Modal Theories", JACM, vol. 29, No. 1.

[Mendelson 1964] Mendelson, E. 1964. *Introduction to Mathematical Logic*, Van Norstrand, Reinhold Co., New York.

[Moore 1985] Moore, R. C. 1985. "Semantical Considerations on Nonmonotonic Logic" *Artificial Intelligence*, 25.

[Parks 1976] Parks, Z. 1976. "An Investigation into Quantified Modal Logic", Studia Logica 35, p109-125.

[Quine 1969] Quine, W.V.O., Set Theory and Its Logic, revised edition, Oxford University Press, London, 1969.

Author information

Frank M. Brown- Artificial Intelligence Laboratory, University of Kansas, Lawrence, Kansas, 66045, e-mail: brown@ku.edu.

ONE APPROACH FOR THE OPTIMIZATION OF ESTIMATES CALCULATING ALGORITHMS

A.A. Dokukin

Abstract: In this article the new approach for optimization of estimations calculating algorithms is suggested. It can be used for finding the correct algorithm of minimal complexity in the context of algebraic approach for pattern recognition

Keywords: Pattern recognition, estimates calculating algorithms.

Introduction

This work is made in the context of algebraic approach [1] (in what follows, we use the notation and definitions from [1,2]) for pattern recognition. The task of recognition is considered. We have a set *M* of possible objects. It is presumed that $M = M_1 \times ... \times M_n$, there M_i are sets of possible values of *i*-th feature, and some semi-metrics are defined on each of them. The set *M* is divided into *l* classes $K_1,...,K_l$. The task of recognition is defined by the conventional learning information $I_0 = \{S_1,...,S_m,\alpha(S_1),...,\alpha(S_m)\}$, and the finite sample, $\beta(S^i) = (\beta_{i1},...,\beta_{il})$ of test objects. Here $S_1,...,S_m$ are descriptions of training sequence objects $S_i = (a_{i1}, a_{i2},...,a_{in}), a_{ij} \in M_j,\quad i = \overline{1,m}, j = \overline{1,n}$, and $\alpha(S_i) = (\alpha_{i1},...,\alpha_{il})$ are information vectors of objects S_i, with respect to the properties $P_j(S) \equiv \{S \in K_j\},\ j = \overline{1,l}$. Correspondently $\beta(S^j) = (\beta_{i1},...,\beta_{il})$ are information vectors of S^j.

The task is to find algorithm in the algebraic closure of some set of recognition operators that calculates information vector for each $S^i \in \widetilde{S}^q$. As such system the defined below class of ECA (estimates calculating algorithm) is considered.

Yu.I. Zhuravlev have proved [1] that there exixts a correct polynomial in the algebraic closure of ECA, i.e. polynomial that provides no errors on the control information $\widetilde{S}^q, \{\beta(S^1),...,\beta(S^q)\}$.

Estimates calculating algorithm A is defined as $A = B \cdot C$, where $B(I_0, \widetilde{S}^q) = \|\Gamma_{ij}\|_{q\times l} = \|\Gamma_j(S^i)\|_{q\times l}$ is recognition operator, $C(\|\Gamma_{ij}\|_{q\times l}) = \|\beta_{ij}\|_{q\times l}$ is solving rule.

$$\Gamma_j(S^i) = x_1\Gamma^1_j(S^i) + x_0\Gamma^0_j(S^i). \tag{1}$$

$$\Gamma^1_j(S^i) = \frac{1}{Q_1}\sum_{S\in\widetilde{K}_j}\sum_{\omega\in\Omega_A}\gamma(S^i)p(\omega)B(\omega S^i, \omega S) \tag{2}$$

$$\Gamma^0_j(S^i) = \frac{1}{Q_0}\sum_{S\in C\widetilde{K}_j}\sum_{\omega\in\Omega_A}\gamma(S^i)p(\omega)\overline{B(\omega S^i, \omega S)}. \tag{3}$$

Following notation is used:

- The *j*-th class and its addition are denoted as $\widetilde{K}_j = K_j \cap \{S_1,...,S_m\}$ and $C\widetilde{K}_j = \{S_1,...,S_m\} \setminus \widetilde{K}_j$.

- Let $\{\Omega\}$ is the set of all subsets of $\{1,...,n\}$. Some subset Ω_A of Ω is attributed to an algorithm. Its elements $\omega_t = \{i_1,...,i_{k_t}\} \in \{\Omega_A\}$ are called support sets and $p(\omega_t) = p_{i1} + ... + p_{ik_t}$ are their weights, $p(\omega_t) \geq 0$.
- $\gamma(S^i) \geq 0$ are weights of training objects.
- $B(\omega S^i, \omega S)$ is proximity function. We use proximity functions only of the following type. Let $\varepsilon_1,...,\varepsilon_n$ are non-negative numbers, let also $\omega S = \{a_{i1},...,a_{ik}\}$, $\omega S' = \{b_{i1},...,b_{ik}\}$ then $B(\omega S, \omega S') = \begin{cases} 1, & \rho_{i1}(a_{i1}, b_{i1}) \leq \varepsilon_{i1},...., \rho_{ik}(a_{ik}, b_{ik}) \leq \varepsilon_{i\varepsilon} \\ & 0, \quad otherwise \end{cases}$.

  $\overline{B(\omega S^i, \omega S)} = 1 - B(\omega S^i, \omega S)$.

Denote a set of recognition operators by $\{\widetilde{B}\}$. Let $B', B'' \in \{\widetilde{B}\}$, $B'(I_0, \widetilde{S}^q) = \left\|\Gamma_{ij}'\right\|_{q\times l}$, $B''(I_0, \widetilde{S}^q) = \left\|\Gamma_{ij}''\right\|_{q\times l}$, *b* is a scalar. Following operations bB', $B' + B''$, $B' \cdot B''$ can be defined on this set as shown below.

$$(bB')(I_0, \widetilde{S}^q) = \left\|b\Gamma_{ij}'\right\|_{q\times l} \tag{4}$$

$$(B' + B'')(I_0, \widetilde{S}^q) = \left\|\Gamma_{ij}' + \Gamma_{ij}''\right\|_{q\times l} \tag{5}$$

$$(B' \cdot B'')(I_0, \widetilde{S}^q) = \left\|\Gamma_{ij}' \cdot \Gamma_{ij}''\right\|_{q\times l} \tag{6}$$

The closure $M(\{\widetilde{B}\})$ with respect to operations (4)-(6) is associative algebra with commutative multiplication. Operators from $M(\{\widetilde{B}\})$ can be presented as polynomials of operators from $\{\widetilde{B}\}$. If $B \in M(\{\widetilde{B}\})$ then $B = \sum B_{i_1} \cdot B_{i_2} \cdot ... \cdot B_{i_t}$. The maximum number of multipliers in its items is called the degree of recognition operator.

The family $M(\{A\})$ of algorithms $A = B \cdot C$ such that $B \in M(\{\widetilde{B}\})$ is called algebraic closure of $\{A\}$.

Finally we will need some more terms from [3] to continue the statement. The informational matrix $\left\|\beta_{i,j}\right\|_{q\times l}$ is considered. Suppose $M = \{(i,j)\}$, $i = 1,...,q$, $j = 1,...,l$, $M_\alpha = \{(i,j) \mid \beta_{i,j} = \alpha\}$, $\alpha \in \{0,1\}$.

Operator $B \in M(\{\widetilde{B}\})$ is called admissible if there exists at least one pair $(i,j) \in M_1$ such that for all pairs $(u,v) \in M_0$ $\Gamma_j(S^i) > \Gamma_v(S^u)$. This pair is called marked. It is proved also [3] that the greater value $d(i,j,B) = \min\limits_{(u,v)\in M_0} (\Gamma_j(S^i) - \Gamma_v(S^u))$ is the smaller degree of item will be needed to construct the correct polynomial.

Thus in order to construct a correct algorithm of minimal complexity or to make inductive procedure of constructing it (like for example one in [4]), we need to find the algorithm of maximum $d(i,j,B)$ in some family of algorithms. This article is devoted to solving of maximization task in two particular subsets of ECA.

γ-optimization

First, denote by $\{B\}_\gamma$ the subset of ECA with the following parameters:

- $x_0 = 0, x_1 = 1$,
- Ω_A consists of all support sets of equal fixed power k. $p_i = 1/k, i = 1,...,n$,
- $\tilde{\gamma} \in [0,1]^m$,
- $\varepsilon_1,...,\varepsilon_n$ are fixed.

Let we have $(i, j) \in M_1$. The task is to find $\tilde{\gamma}^* \in [0,1]^m$ such that

$$\max_{B\in\{B\}_\gamma} \min_{(u,v)\in M_0} (\Gamma_j(S^i) - \Gamma_v(S^u)) = \min_{(u,v)\in M_0} (\Gamma_j(S^i) - \Gamma_v(S^u)) |_{\tilde{\gamma}=\tilde{\gamma}^*} . \tag{7}$$

As shown in [1], in case of this special format of support vectors, the estimations (1)-(3) can be transformed into simple view:

$$\Gamma_j(S^i) = x_1\Gamma^1_j(S^i) + x_0\Gamma^0_j(S^i) = \Gamma^1_j(S^i),$$

$$\Gamma^1_j(S^i) = \frac{1}{Q_1} \sum_{S\in\tilde{K}_j} \sum_{\omega\in\Omega_A} \gamma(S^i) p(\omega) B(\omega S^i, \omega S) =$$

$$= \frac{1}{Q_1} \sum_{S\in\tilde{K}_j} \gamma(S)((\delta(S,S^i)\cdot p(\omega))V^1 + (\tilde{\delta}(S,S^i)\cdot p(\omega))V^0)$$

Here $\delta(S,S^i) \in \{0,1\}^n$ is the characteristic vector $\delta_u((a_1,...,a_n),(b_1,...,b_n)) = \begin{cases} 1, \rho_u(a_u,b_u) \le \varepsilon_u \\ 0, \rho_u(a_u,b_u) > \varepsilon_u \end{cases}$,

$\tilde{\delta}(S,S^i) \in \{0,1\}^n$ is its denial : $\tilde{\delta}_u = 1 - \delta_u$, $q(S,S^i) = \sum_{u=0}^{n} \delta_u$.

$$V^1(S,S^i) = \sum_{u=0}^{\varepsilon} C^u_{n-q(S,S^i)} C^{k-u-1}_{q(S,S^i)-1},\ V^0(S,S^i) = \sum_{u=1}^{\varepsilon} C^{u-1}_{n-q(S,S^i)-1} C^{k-u}_{q(S,S^i)} .$$

So in the $\{B\}_\gamma$ family of ECA, the estimation $\Gamma_j(S^i) - \Gamma_v(S^u)$ is linear function on $\tilde{\gamma} \in [0,1]^m$, that is $\Gamma_j(S^i) - \Gamma_v(S^u) = L_{i,j,u,v}(\tilde{\gamma})$. So the task transforms into another one, i.e. to find $\arg\max_{\tilde{\gamma}\in\tilde{\gamma}^*} \min_{(u,v)\in M_0} L_{i,j,u,v}(\tilde{\gamma})$, there $L_{i,j,u,v}(\tilde{\gamma}) = \sum_{s=1}^{m} l_{s,i,j,u,v}\gamma_s$. This task in turn can be transformed into *t* tasks of linear programming, there $t = |M_0|$ (we enumerate all those linear combinations as $L_1,...,L_t$ in any order):

$$\tilde{\gamma}^* = \arg\max_{\tilde{\gamma}\in\{\tilde{\gamma}_1^*,...,\tilde{\gamma}_t^*\}} L_i(\tilde{\gamma}_i^*)$$

$$\begin{cases} \tilde{\gamma}_i^* = \arg\max_{\tilde{\gamma}} L_i(\tilde{\gamma}) \\ L_i(\tilde{\gamma}) \le L_1(\tilde{\gamma}) \\ ... \\ L_i(\tilde{\gamma}) \le L_t(\tilde{\gamma}) \\ \tilde{\gamma} \in [0,1]^m \end{cases} ,\ i = 1,...,t .$$

These tasks can be solved with, for example, simplex method. So the precise solution of the initial task can be found.

γ,ε-optimization

The second task is more complex. As in previous chapter we choose parametrical subset $\{B\}_{\gamma,\varepsilon}$ of ECA first:

- $x_0 = 0, x_1 = 1$,
- Ω_A consists of the single support set (the method can be simply generalized to include cases of small number of support sets),
- $\tilde{\gamma} \in [0,1]^m$,
- $\varepsilon_1,...,\varepsilon_n \geq 0$.

The task is the same as in previous section, i.e. to find in $\{B\}_{\gamma,\varepsilon}$ the algorithm with the maximum value of $d(i,j,B)$.

The algorithm for solving of this task consists of two parts. First one is the construction of auxiliary finite system of parallelepipeds P:

1. Build new sequence of objects $\{S'_1,...,S'_t\}$: for all $S \in \tilde{K}_j$ add differences $S^i - S$ to the sequence.
2. Find the minimal system P of parallelepipeds $[-\varepsilon_1,\varepsilon_1]\times...\times[-\varepsilon_n,\varepsilon_n]$ containing all different combinations of objects from $\{S'_1,...,S'_t\}$.

To construct the system P we must for all subsets $S \subset \{S'_1,...,S'_t\}$ find out if its combination is possible, i.e. if there exists any parallelepiped $E = [-\varepsilon_1,\varepsilon_1]\times...\times[-\varepsilon_n,\varepsilon_n]$ such that $S' \in S$ if and only if $S' \in E$, and for all possible combinations add the minimal parallelepiped spanning it to the system. In practice there is no need to enumerate all different subsets of $\{S'_1,...,S'_t\}$. If we have found any impossible one, every combination containing it is impossible too.

The following theorem can be proved: $\max\limits_{\varepsilon\in[0,\infty)^n} \min\limits_{(u,v)\in M_0} (\Gamma_j(S^i) - \Gamma_v(S^u)) = \max\limits_{\varepsilon\in P} \min\limits_{(u,v)\in M_0} (\Gamma_j(S^i) - \Gamma_v(S^u))$. Indeed for any ε-neighborhood $[-\varepsilon_1,\varepsilon_1]\times...\times[-\varepsilon_n,\varepsilon_n]$ the maximum one from P containing in it will give not more estimations.

The second part is to calculate estimations themselves and solve the task. From (1)-(3) we have

$\Gamma_j(S^i) = g_1\gamma_1 + ... + g_n\gamma_n, g_k \in \{0,1\}, k = 1,...,n$,

$\Gamma_v(S^u) = g_1^{u,v}\gamma_1 + ... + g_n^{u,v}\gamma_n, g_s^{u,v} \in \{0,1\}, s = 1,...,n$.

And the difference is

$$\Gamma_j(S^i) - \Gamma_v(S^u) = g_1\gamma_1 + ... + g_n\gamma_n - g_1^{u,v}\gamma_1 - ... - g_n^{u,v}\gamma_n \quad (8)$$

So the solution is

$$\tilde{\gamma}^* : \gamma_s^* = \begin{cases} 1, & g_s = 1 \\ 0, & otherwise \end{cases}, \quad s = 1,...,n.$$

Indeed for any $\tilde{\gamma} \in [0,1]^n$, difference (8) is smaller than $\Gamma_j^*(S^i) - \Gamma_v^*(S^u) = g_1\gamma_1^* + ... + g_n\gamma_n^* - g_1^{u,v}\gamma_1^* - ... - g_n^{u,v}\gamma_n^*$. The initial task transforms into finding $\arg\max\limits_{\varepsilon\in P} \min\limits_{(u,v)\in M_0} \Gamma_j(S^i) - \Gamma_v(S^u)$ and the precise solution can be found too.

Though the solution is precise the necessity to construct system P makes the task extremely difficult with multidimensional data. In order to make calculation faster we suggest proximate method for the same task.

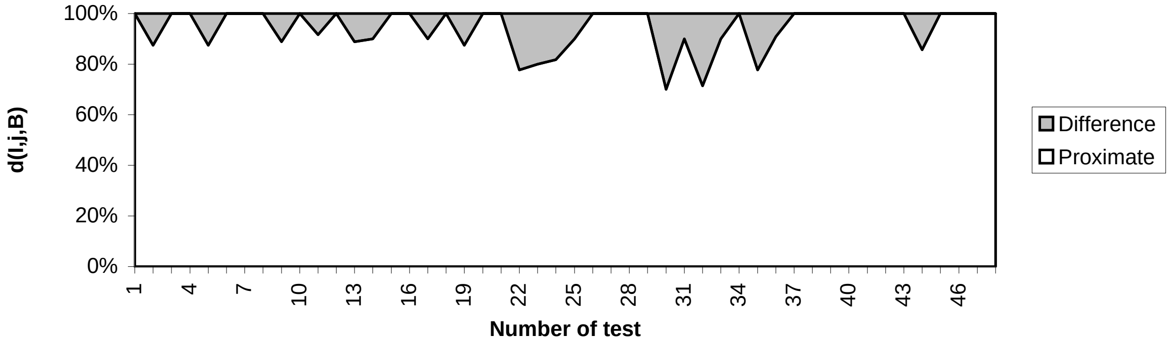
The method starts with the parallelepiped spanning the whole sequence $\{S'_1,...,S'_t\}$. Then on every step we enumerate all admissible combinations of t-1 objects and leave the best one for next step, there we consider neighborhood spanning those best combination. Here t is the number of objects in current parallelepiped. The best combination is one that maximizes the value of $d(i,j,B)$.

The following diagram shows results of hands-on testing of this method in comparison with the precise one. The table of descriptions of forty-eight patients was considered. It consists of three classes of correspondingly seventeen, twenty and twelve objects and thirty-three features. As the M_1 in turns every object was considered. All other objects from its class were considered as the training sequence. All objects from other classes formed M_0. For example the twentieth object generated the following (20-th) test:

$$M_1 = \{(20,2)\}$$

$$M_0 = \{(1,2),(2,2),...,(17,2),(38,2),(39,2),...,(48,2)\}$$

$$\{S'_1,...,S'_t\} = \{S_{18}, S_{19}, S_{21}, S_{22},..., S_{37}\}.$$



It's easy to see that in most cases the precise solution or solution of acceptable precision has been found. And while the precise solution takes about two minutes to find (in case of twenty training objects and the difficulty extremely grows with increasing of their number), the proximate algorithm performs all forty-eight tests within about ten seconds.

Conclusion

In this article we have suggested the new approach for optimization of estimations calculating algorithms. It can be used for finding of the correct algorithm of the minimal complexity in the context of the algebraic approach for the pattern recognition.

Also we have considered two parametrical subsets of ECA and have found precise algorithms for solving optimization task for them.

Finally the fast proximate method with acceptable precision has been suggested.

Acknowledgements

The research described in this publication was made possible as a part of Grants 02-01-00558, 00-01-00650, 02-07-90134, 02-07-90137 from the Russian Fund of Fundamental Research, and INTAS 00-650, INTAS 00-370.

Bibliography

[1] Yu.I.Zhuravlev. Ob algebraicheskom podhode k resheniyu zadach raspoznavaniya ili klassifikacii // Problemy kibernetiki. 33, M: Nauka, 1978. (in russian)

[2] Yu.I.Zhuravlev. Korrektnye algebry nad mnozhestvom nekorrektnyh (evristicheskih) algoritmov // Kibernetika. 1977. 4. (in russian)

[3] Yu.I.Zhuravlev, I.V. Isaev. Postroenie algoritmov rasposnavania, korrektnyh dlya zadannoi kontrol'noi vyborki // Zhurnal vychislitel'noi matematiki I matematicheskoi fiziki, t.19, N3, may-june1979. (in russian)

[4] A.A.Dokukin. Induktivnyi metod postroeniya korrektnogo algoritma v algebrah nad modelyu vychisleniya ocenok // Zhurnal vychislitel'noi matematiki I matematicheskoi fiziki, 2003. (in russian)

Author information

Alexander A. Dokukin – Dorodnicyn Computing Centre of the Russian Academy of Sciences, Vavilov st., 40, Moscow GSP-1, 119991, Russia; e-mail: dalex@ccas.ru

WEB-BASED SIMULTANEOUS EQUATION SOLVER[12]

A. Iliev, N. Kyurkchiev, T. Todorov

Abstract: In this paper we present methods, theoretical basis of algorithms, and computer tools, which we have used for constructing our Web-based equation solver.

Keywords: automatic equation solver, Web interface, simultaneous extraction of all roots, simultaneous methods, parallel processors, algebraic equations

2000 Mathematics Subject Classification: 68Q22, 65Y05

Introduction

Many industrial and optimization tasks lead to the problem of finding all roots of (1) or arbitrary their part. One of branches for solving polynomial equations is parallel methods for simultaneous determination of all roots. With our solver automatically we can search simultaneously all or only one part of all roots of (1) (real, complex, lying in given area).

Iteration methods

Let us consider algebraic polynomial

$$\begin{aligned} A_n(x) &= x^n + a_1 x^{n-1} + a_2 x^{n-2} + \ldots + a_n = \\ &= \left(\ldots\left(\left(x + a_1\right)x + a_2\right)x + \ldots + a_{n-1}\right)x + a_n . \end{aligned} \tag{1}$$

The approximations of the k th iteration to zeroes $x_1, x_2, \ldots, x_w$ of (1) are denoted by $x_1^{[k]}, x_2^{[k]}, \ldots, x_w^{[k]}$ and their multiplicities by

$$\alpha_1, \alpha_2, \ldots, \alpha_w \ \left(1 \le \alpha_i \le n - w + 1, i = \overline{1, w}, \sum_{i=1}^{w} \alpha_i = n\right).$$

Classical methods for individual searching of multiple roots of (1) can be written in this general way

$$\begin{aligned} & x_i^{[k+1]} = x_i^{[k]} - F\left(x_i^{[k]}, \alpha_i, a_1, a_2, \ldots, a_n\right), \\ & \quad i = \overline{1, w}\,, \ \ k = 0,1,2,\ldots . \end{aligned} \tag{2}$$

Other approach is given by methods for simultaneous extraction of all multiple roots and we can write them as

$$\begin{aligned} & x_i^{[k+1]} = x_i^{[k]} - F\left(x_1^{[k]}, x_2^{[k]}, \ldots, x_w^{[k]}, \alpha_1, \alpha_2, \ldots, \alpha_w, a_1, a_2, \ldots, a_n\right), \\ & \quad i = \overline{1, w}\,, \ \ k = 0,1,2,\ldots . \end{aligned} \tag{3}$$

Methods (3) are steadier and also they have a larger domain of convergence with comparison with methods (2). This is the main reason because these methods are object of detailed investigations in last twenty years.

For natural reasons we want to find simultaneously only one part of all roots of (1). Namely, we want to find simultaneously $p(\le w)$ different roots with multiplicities $\alpha_1, \alpha_2, \ldots, \alpha_p$

[12] This work has been supported by NIMP, University of Plovdiv under contract No MU-1.

$\left(\alpha_1+\alpha_2+\ldots+\alpha_p=n-m,n-m-p+1\geq\alpha_j\geq 1,j=\overline{1,p}\right)$. In our Web-based equation solver we use [Iliev, Kyurkchiev, 2002a, 2002b, 2003], [Kyurkchiev, Iliev, 2002] type methods, which in common can be written as

$$x_i^{[k+1]}=x_i^{[k]}-F\left(x_1^{[k]},x_2^{[k]},\ldots,x_p^{[k]},\alpha_1,\alpha_2,\ldots,\alpha_p,a_1,a_2,\ldots,a_n\right),\quad i=\overline{1,p}\,,\ k=0,1,2,\ldots. \tag{4}$$

Theoretical basis of our methods

Polynomial (1) can be presented in this way

$$A_n(x)=Q_{n-m}(x)T_m(x)\,, \tag{5}$$

where $Q_{n-m}(x)$ is polynomial, whose zeroes we seek and $T_m(x)$ is polynomial, whose zeroes we ignore. Respectively

$$Q_{n-m}(x)=x^{n-m}+b_1x^{n-m-1}+b_2x^{n-m-2}+\ldots+b_sx^{n-m-s}+\ldots+b_{n-m}$$
$$T_m(x)=x^m+c_1x^{m-1}+c_2x^{m-2}+\ldots+c_px^{m-p}+\ldots+c_m.$$

Between the coefficients of polynomial (1) and the coefficients of polynomials (5) there exist the following relations

$$\begin{aligned}
&a_1=c_1+b_1\\
&a_2=c_2+b_2+c_1b_1\\
&\ldots\\
&a_l=c_l+b_l+c_1b_{l-1}+c_2b_{l-2}+\ldots+c_{l-1}b_1\\
&\ldots\\
&a_m=c_m+b_m+c_1b_{m-1}+c_2b_{m-2}+\ldots+c_{m-1}b_1\\
&a_{m+1}=c_2b_{m-1}+c_3b_{m-2}+\ldots+c_mb_1+b_{m+1}\\
&\ldots\\
&a_n=c_mb_{n-m}.
\end{aligned}$$

We want to find simultaneously $p(\leq w)$ different roots with multiplicities

$$\alpha_1,\alpha_2,\ldots,\alpha_p\ \left(\alpha_1+\alpha_2+\ldots+\alpha_p=n-m,n-m-p+1\geq\alpha_j\geq 1,j=\overline{1,p}\right)$$

and we set

$$\begin{aligned}
Q_{n-m}^{[k]}(x)&=\prod_{j=1}^{p}\left(x-x_j^{[k]}\right)^{\alpha_j}=\\
&=x^{n-m}+b_1^{[k]}x^{n-m-1}+b_2^{[k]}x^{n-m-2}+\ldots+b_s^{[k]}x^{n-m-s}+\ldots+b_{n-m}^{[k]},\\
T_m^{[k]}(x)&=x^m+c_1^{[k]}x^{m-1}+c_2^{[k]}x^{m-2}+\ldots+c_p^{[k]}x^{m-p}+\ldots+c_m^{[k]}.
\end{aligned} \tag{6}$$

From (6) it follows

$$\begin{aligned}
&b_1^{[k]}=-\sum_{j=1}^{p}\alpha_jx_j^{[k]}\\
&b_2^{[k]}=\sum_{j=1}^{p-1}\left[\alpha_jx_j^{[k]}\sum_{s=j+1}^{p}\alpha_sx_s^{[k]}\right]+\sum_{j=1}^{p}\frac{\alpha_j\left(\alpha_j-1\right)}{2}\left(x_j^{[k]}\right)^2\\
&\ldots\\
&b_{n-m}^{[k]}=(-1)^{n-m}\prod_{j=1}^{p}\left(x_j^{[k]}\right)^{\alpha_j}\,.
\end{aligned}$$

Combinative algorithms can be used for finding coefficients $b_1^{[k]}, b_2^{[k]}, \dots, b_{n-m}^{[k]}$.

We define $c_j^{[k]}, \, j = \overline{1,m}$ using formulae

$$c_1^{[k]} = a_1 - b_1^{[k]}$$
$$c_2^{[k]} = a_2 - b_2^{[k]} - \left(a_1 - b_1^{[k]}\right)b_1^{[k]} = a_2 - b_2^{[k]} - c_1^{[k]}b_1^{[k]}$$
$$\dots$$
$$c_m^{[k]} = F(a_1, a_2, \dots, a_m, b_1^{[k]}, b_2^{[k]}, \dots, b_m^{[k]}) = a_m - b_m^{[k]} - \sum_{j=1}^{m-1} c_j^{[k]} b_{m-j}^{[k]}.$$

For simultaneous searching of roots of $Q_{n-m}(x)$ from (5) we give the following iteration algorithm

$$x_i^{[k+1]} = x_i^{[k]} - \frac{\alpha_i A_n\left(x_i^{[k]}\right)}{A_n'\left(x_i^{[k]}\right) - A_n\left(x_i^{[k]}\right)\left[\sum_{j=1, j\neq i}^{p} \frac{\alpha_j}{x_i^{[k]} - x_j^{[k]}} + \frac{T_m'^{[k]}\left(x_i^{[k]}\right)}{T_m^{[k]}\left(x_i^{[k]}\right)}\right]}, \qquad (7)$$

$i = \overline{1,p}$, $k = 0,1,2,\dots$.

When $m = n-1$ and $\alpha_1 = \alpha_2 = \dots = \alpha_n = 1$ method (7) coincides with the classical Obreshkoff's method [Obreshkoff, 1963] for individual searching of one simple zero and if $p = w$ (7) is method for finding all roots of (1).

Theorem. Let $d \overset{\text{def}}{=} \min_{i \neq j} \left|x_i - x_j\right|$, $c > 0$ and $1 > q > 0$ be real numbers such that

$$d > 2c,$$
$$2c^2\left[\left(n - m - \alpha_i\right)/(d - 2c)^2 + \left(gP_2 + yP_1\right)G_1^{-2}\right] < \alpha_i, \ i = \overline{1,p},$$

where P_1, P_2, G_1, g and y are appropriate positive constants. If initial approximations $x_1^{[0]}, x_2^{[0]}, \dots, x_p^{[0]}$ to the real roots of (1) satisfy inequalities $\left|x_i^{[0]} - x_i\right| < cq, i = \overline{1,p}$ then for every $k \in N$ the inequalities

$$\left|x_i^{[k]} - x_i\right| < cq^{3^k}, i = \overline{1,p}$$

are satisfied.

From this theorem [Iliev, Kyurkchiev, 2003] it follows that iteration method (7) holds cubic convergence.

Localization technique for automatic determination of multiplicity of the roots and their initial approximations

For applying in practice in Obreshkoff's monograph [Obreshkoff, 1963] is given that Fujiwara prove that the circle with centre origin and radius $R = 2\max_{1 \le p \le n}\left|a_{n-p} / a_n\right|^{1/p}$ in the complex plane contains all zeroes of polynomial (1).

We will use presentation

$$A_n(x) = \left(x - x_1\right)\left(x - x_2\right)\dots\left(x - x_n\right) = \rho_1\rho_2\dots\rho_n e^{i\varphi_1}e^{i\varphi_2}\dots e^{i\varphi_n},$$

where ρ_p are modulo of complex numbers $x - x_p$, $p = \overline{1,n}$ and φ_p, $p = \overline{1,n}$ are their arguments.

After one pass of contour with appropriate step in counter-clockwise direction every arguments of the roots in domain will be changed with 2π and every arguments out of contour will not be changed. Using Cauchy approach [Obreshkoff, 1963] if the argument change of $z = A_n(x)$, where x with appropriate step in counter-clockwise direction are different points from passed contour, is $2\pi s$, $s \in [1, n]$, it follows that in explored domain there are exactly s roots.

After first pass of localization of the roots with presented here algorithm in different domains we explore these domains, which have more than one root. For every such area arises the question whether or not in it there are localized one or several roots, which are sufficiently close. Confirmation or rejection of found multiplicity in "near" neighborhood could be made with Schröder's method [Schröder, 1870]. It will have quadratic convergence only when the multiplicity of the root is exact. Exactly we have in mind ε discernible roots (zeroes). If the roots are not multiple we will repeat Cauchy algorithm procedure. This is because we need fine localization of roots only in these areas, which contain more than one different roots.

Program description

The main modules are realized on Pascal program language [Krushkov, Iliev, 2002], using Delphi 5 environment. We realized specialized program units for complex numbers, multiple precision, input polynomial analyzer, specialized methods for finding a part of all roots [Iliev, Kyurkchiev, 2002a, 2002b, 2003], [Kyurkchiev, Iliev, 2002]. Also we have used dynamic structures for the economy of memory and for faster program code optimization. For Web-based input-output form interface for users is developed.

Conclusion

With this equation solver we try to give practical cover of theoretical improvements from classic, advanced techniques and iteration algorithms from last several years. It can be used simultaneously from many users with Internet connection without influence of distance.

Bibliography

[Iliev, Kyurkchiev, 2002a] A. Iliev, N. Kyurkchiev. On a Generalization of Weierstrass-Dochev Method for Simultaneous Extraction of only a Part of All Roots of Algebraic Polynomials: Part I. C. R. Acad. Bulg. Sci., 2002, 55, № 2, 23–26.

[Iliev, Kyurkchiev, 2002b] A. Iliev, N. Kyurkchiev. Some Methods for Simultaneous Extraction of only a Part of all Roots of Algebraic Polynomials Part II. C. R. Acad. Bulg. Sci., 2002, 55, № 9, 17–22.

[Kyurkchiev, Iliev, 2002] N. Kyurkchiev, A. Iliev. On the *R*-order of Convergence of a Family of Methods for Simultaneous Extraction of Part of All Roots of Algebraic Polynomials. BIT, 2002, 42, № 4, 879–885.

[Iliev, Kyurkchiev, 2003] A. Iliev, N. Kyurkchiev. On a Generalization of Obreshkoff-Ehrlich Method for Simultaneous Inclusion of only Part of All Multiple Roots of Algebraic Polynomials. Comput. Math. Math. Phys., 2003, 43, № 4, 507–513.

[Obreshkoff, 1963] N. Obreshkoff. On the numerical solving equations. Ann. of the Sofia University, 1963, 56, 73–83.

[Obreshkoff, 1963] N. Obreshkoff. Polynomial zeroes. Acad. Bulg. Sci., Sofia, 1963.

[Schröder, 1870] E. Schröder. Über unendlich viele Algorithmen zur Auflösung der Gleichungen. Math. Ann. 2, 1870, 317–365.

[Krushkov, Iliev, 2002] H. Krushkov, A. Iliev. Practical programming guide in Pascal, Parts I and II. Anima, 2002.

Authors information

Anton Iliev - University of Plovdiv, Faculty of Mathematics and Informatics, Department of Computer Science, 24, Tsar Assen Str., Plovdiv-4000, Bulgaria; e-mail: aii@pu.acad.bg, http://anton.iliev.tripod.com

Nikolay Kyurkchiev - Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, Acad. G. Bonchev Str., bl. 8, Sofia-1113, Bulgaria; e-mail: nkyurk@math.bas.bg

Todor Todorov - University of Plovdiv "Paisii Hilendarski", Faculty of Mathematics and Informatics, Department of Computer Science, 24, Tsar Assen Str., Plovdiv-4000, Bulgaria; e-mail: tod_bul@hotmail.com

VIRTUAL INSTRUMENTS – FUNCTIONAL MODEL, ORGANIZATION AND PROGRAMMING ARCHITECTURE

G.S.Georgiev, G.T.Georgiev, S.L.Stefanova

Abstract: This paper presents functional model, organization, a programming architecture and an implementation of Virtual Instruments as an essential part of educational laboratory tools. The Virtual Instruments are designed in event- driven programming environment and are capable of performing instrumental functions in local or remote level. The possibility of realization of real time operations from signal information point of view is discussed.

Key words: Local Virtual Instrument, Remote Virtual Instrument, Events, Messages, DAQ System, DLL, Sockets, Java RMI

Introduction

Modern measurement systems for data acquisition and processing in engineering and research combine three basic functions:

- DATA ACQUISITION. Usually this comprises a number of measured quantities, characterising the behaviour of the object of measurement; they are sampled simultaneously or sequentially, in most cases multiplexing the measured signals through several analogue channels (8, 16, 32 or more) for further conversion by a common analogue to digital converter. This function is implemented in hardware by DAQ-systems (DATA ACQUISITION SYSTEMS);
- DATA ANALYSIS by means of algorithms for processing the results from multiple, aggregate or combined measurements and specific procedures for measurement and calculation which eliminate systematic errors, depending on the measured quantity and the environment within which it is monitored. Usually this function requires performing a large amount of computational and logical operations for reducing the initial indetermination of the quantity under measurement by comparing it with what is called best value and interval of residual indetermination - standard deviation;
- DATA PRESENTATION. Most often this comprises visual relations among measured data in graphical or table form. Their suitable visualisation has a certain (often decisive) impact on the quality of the carried out measurement process. Usually when conducting engineering or scientific research one has to "experiment" with scaling of graphs, approximation of the processed signals and visualisation of the functional relations.

The implementation of general informational and specific measurement procedures determine the efficiency of the means for carrying out a measurement process and the opportunity for achieving its goals. Virtual instruments [1], as a combination of (quick) hardware and (flexible) software can be part of a well defined teaching hierarchical structure for:

- ➢ generation of asynchronous data streams for physical phenomena and properties in the object of measurement;
- ➢ classification, discovering of interrelations among them and merging them into a common data base;
- ➢ processing of the merged data in order to represent it in accordance with the objectives of the knowledge extraction process and to allow for interaction with the consumer of knowledge.

Building the measuring instruments into a suitable computer environment can carry out efficient accomplishment of these functions. In such a structure, which is new in a qualitative aspect, the computer environment controls the conversion process for efficient performance of analysis and visualisation. So developed computer based instruments can be helpful in advancing of teaching process [1]. In order to promote not only the theoretical but also the practical work of learners in contemporary computer based laboratories there is a trend to integrate the basic theoretical material with simulation, animation, quizzes and practical experiments. Such an approach, in which the experimenting work of learners brings to virtual instruments, is realized in [7]. The applications developed by Asymetrix Toolbook, Microsoft Visual C^{++} in Microsoft Windows operational environment enable a simulation of measuring procedures. The connections with the laboratory objects – copies of the real circuits, devices and systems the learner is working with in his future professional area are missing. A step forward is the expansion of the standard computer configuration with an instrumental hardware. In this direction an application for investigating microprocessor control of a step motor is described in [8]. A disadvantage is the object dependant hardware and controlling primitives that make the expansion of the system impossible. More effective from technical and technological point of view is the approach, proposed by Ponta in [9,10]. The main idea is the separation and standardization of the input/output operation in order to simplify the operations fulfilled by the computer. The hierarchical software structure is more flexible in its adaptation to the different laboratory experiments. Unfortunately the possibility for the user to change the functions of the instruments is missing. The user functions and advantages of Virtual instruments, being come to light in [1], are shown how to be realized through standard ways in [2]. First of all the described approaches concern their local realization, that is not enough for the purposes of a Remote Virtual Laboratory building.

This paper aims to discuss the organization of and approaches for implementing virtual instruments for educational purposes in the structure of a remote teaching virtual laboratory. The presented approaches are generic enough to be used in other areas, too.

Generic organization and functional model

The nature of a measurement process is consistent with generation of messages within the measuring environment, caused by events, reflecting changes in the object of measurement [4]. This makes it natural for the measurement process to be embedded in event-driven software environments, such as Windows, where influences upon the computer environment are caused by events through generation of messages. They can be caused by the user via his/her interactions with graphical elements of the user interface by means of keyboard or mouse. The interactions between events coming from the measurement environment and the operator, by means of messages raised by them, make them "equal in rights" thus making the operator an active participant in the measurement process (Fig. 1).

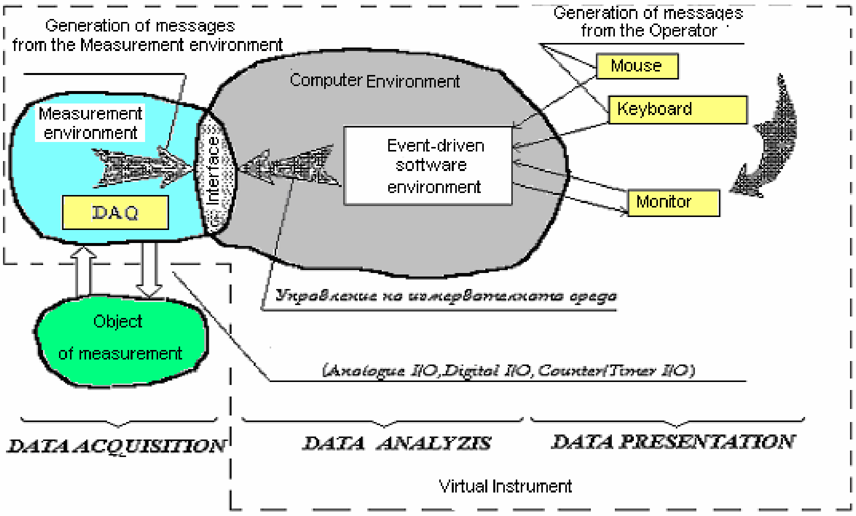


Fig. 1. Functional Model of the Virtual Instrument

The hardware resources of the virtual instrument are defined by the signals for interaction between the DAQ-system and the object of measurement. They can be divided into the following categories: analogue inputs; analogue outputs; digital inputs; digital outputs; timer/counter inputs/outputs.

One essential advantage of virtual instruments is the generalization of their structure by means of a common model for representing the signals. Since the environment, processing the signals as defined by the computer which manages events interaction, is discrete, the relation between the analogue input and output signals and the model for their discrete representation is of decisive importance.

The continuous signal is converted into discrete form, processed in the latter and then, if necessary, converted back to a continuous signal (Fig. 2). In this case the processing of the discrete signal can be done by a general purpose computer, varying the signal processing algorithms while maintaining the general technical structure [2]. Under certain conditions the heterogeneous combination of continuous with discrete signals can be equivalent to a common continuous, time invariant system with frequency response H$_c$(ω), despite the non-time invariants of the pulse modulator [3].

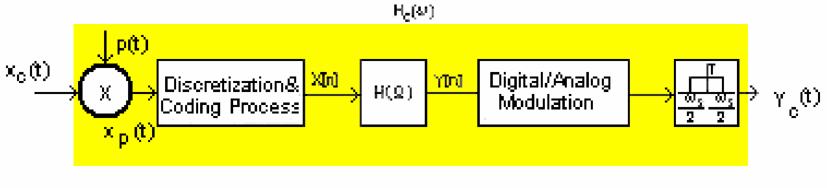


Fig.2 Generic diagram for processing of a continuous signal in a Virtual Instrument

Clarifying the conditions for equivalence is of a crucial importance for correct from an informational point of view carrying out of the measurement process in virtual measurement systems. Modulating the analogue signal x$_c$(t) with a pulse train p(t) with sample period T and sample rate ω_s, $T = 2\pi / \omega_s$, is the most frequently used conversion method:

$$x_p(t) = x_c(t)p(t), \quad p(t) = \sum_{n=-\infty}^{+\infty} \delta(t - nT) \qquad (1)$$

The signal x$_p$(t) is a pulse sequence with pulse amplitudes, equaling the samples from x$_c$(t), in moments displaced by T from each other, and frequency equivalent, determined by the convolution of $X_c(\omega)$ and $P(\omega)$:

$$X_p(\omega) = \frac{1}{2\pi}[X_c(\omega) * P(\omega)] = \frac{1}{T}\sum_{k=-\infty}^{+\infty} X_c(\omega - k\omega_s) \qquad (2)$$

Consequently, $X_c(\omega)$ is a periodic function of frequency and accounts for displaced copies of $X_c(\omega)$, scaled by *1/T*. Restoring the original signal with a limited spectrum ω_M is only possible if displaced copies don't overlap, i.e. $\omega_s \geq 2\omega_M$. Under this condition, the relation between the frequency spectrum $X_c(\omega)$ and the signal $x_c(t)$ is:

$$x_c(t) = \frac{1}{2\pi}\int_{-\omega_M}^{+\omega_M} X_c(\omega) e^{j\omega t} d\omega \qquad (3)$$

Decomposing $X_c(\omega)$ into a Fourier transformation, the following holds true:

$$x_c(t) = \frac{1}{4\pi f_M}\sum_{n=-\infty}^{\infty} x_c(\frac{n}{2f_M})\int_{-2\pi f_M}^{2\pi f_M} e^{j\omega(t - \frac{n}{2f_M})} d\omega = \sum_{n=-\infty}^{\infty} x_c(\frac{n}{2f_M})\frac{\sin(2\pi f_M t - n\pi)}{2\pi f_M t - n\pi} \quad (4)$$

Equation (4) shows that the analogue signal, defined by samples with frequency f_s, can be restored by passing the spectrum of the discrete signal through an ideal low-pass filter with cutting frequency ω_c and adhering to the more generic limiting conditions:

$$\omega_M < \omega_c < \omega_s - \omega_M, \omega_s > 2\omega_M \qquad (5)$$

Failing to observe (5) leads to low-level noise from overlapping of displaced copies' spectrums, also known as *aliasing noise.* This is why incorrect choice of sample frequency leads to unavoidable degradation of the restored signal. Fighting *aliasing noise* should be carried out by precursory limitation of the informative signal's spectrum. For finite in time signals this inevitably leads to the advent of error, which must carefully be considered in preliminary analysis and accounted for in the total balance of errors.

Real-time measurement and preserving of the full information of signals across conversions can be achieved through implementation of local virtual instruments by combining hardware with suitable parameters in the technical part of the measurement system and well considered organization of algorithmical and software resources. In implementing remote virtual instruments, the possibilities for scaling of time are sharply reduced by the time it takes to carry out the routed Internet access, and consequently reduced are the possibilities for carrying out experiments in real-time for most processes under investigation. This is the reason why precursory processing of initially measured data is needed in "immediate proximity" to the object of measurement and integrated forms of estimations should be sent to the client, which, due to generalization, have lost some of the information contained in the measured signals, but have a considerably slower rate of change. They however can be transmitted with suitable for the Internet medium informational frequencies. It is desirable that remote experiments in remotely accessed labs be described by such integrated ratings, so that the trainee, instead of being a passive observer and recorder can turn to a participant in an interactive monitoring, requiring his or her active share in the experiment, consequently better achieving educational goals.

Software implementation

One possible architecture of a virtual instrument is presented on Fig.3.

The system is expected to run under Windows, due to the well-documented methods for accessing the hardware in this operating system. Windows imposes the need for a specific module, kernel mode driver, which is the only module that can access hardware directly, e.g. read from / write to IO ports, react to interrupt requests, etc. In this case it incorporates a number of functions, which are used to read data from ADCs, set values for DACs, start timers, etc. The DLL is in fact a wrapper, which makes the functions, implemented by the kernel mode driver available for use by applications. Additionally, as illustrated on Fig. 3, the DLL can also provide:

- a server socket, implementing the server side of a socket based client-server protocol;
- function(s) that can be called via Java RMI;
- Windows-specific synchronization mechanisms for eliminating conflicts if more than one application tries to access the hardware in any one moment.

The former two allow for remote access to the instrument, while the latter allows multiuser access.

This architecture is based on the event-driven paradigm used by Windows and other graphical user interface systems. The system reacts to events by forming messages, which are then placed in affected applications' message queues. These messages are dispatched by a system call, finally reaching a window function - WndProc on Fig. 3. Control is then passed to specific code, processing any message of interest. As mentioned above, events can enter the system as a result of user interaction (from the keyboard, mouse, etc.). The same message-based approach can be used for asynchronous data exchange with the hardware. This allows a remote user to get integrated forms of estimations, discussed in the previous section. If the hardware allows it, the user can request a series of samples to be taken at a given rate. This request is processed in its entirety locally, even strictly at the hardware level. When the required number of samples has been collected, the hardware generates an interrupt request. This is detected by the kernel mode driver, which sends a specific message to the Windows kernel. This message is then dispatched to any application, which cares to process it. Thus the application receives the whole set of samples, taken in real time, and can visualize them locally, or send them to a remote user via RMI or sockets.

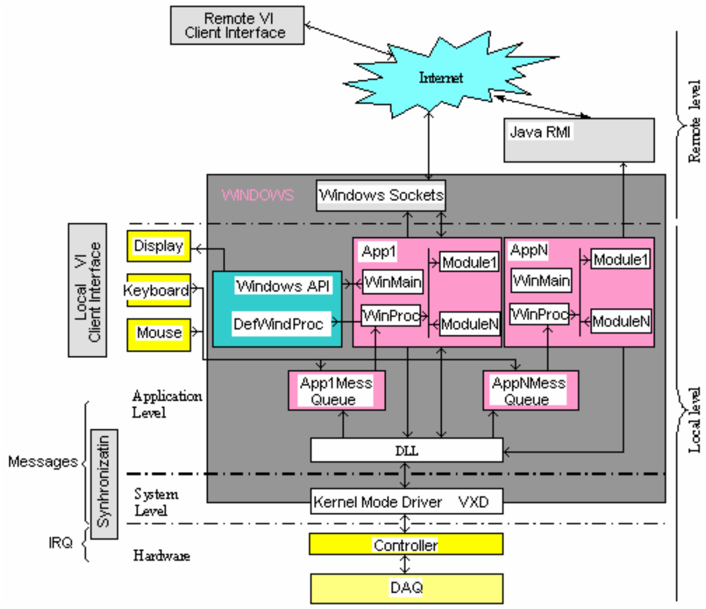


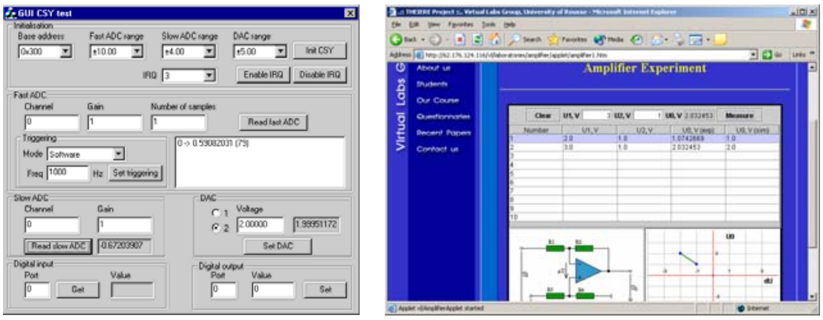
Fig. 3 Software architecture of a virtual instrument

Another important feature of this architecture is that all requests for hardware access, whether local or remote, rely on one module, the DLL, for fullfilment of these requests. This makes it very easy to implement synchronization, using standard Windows synchronization objects, such as critical sections or semaphores. Furthermore, if the hardware allows it, the different devices within it (such as DACs, ADCs, etc.) can be virtualised seperately and allow concurrent access of more than one user to different devices. The only requirement to the hardware for achieving such functionality is that it generates IRQ for each individual device and has means to identify which device caused it.

Realization and conclusion

Using above discussed technology a local test Virtual instrument (fig.4a) for functions control of a DAQ system and a Remote Virtual instrument for analysis of input/output characteristics of Instrumentation amplifiers (fig 4b) are developed. Both of them are implemented as powerful tools in the structure of the Remote Virtual Lab in Rousse University (http://tie.ru.acad.bg) [5]. The first is used by teaching staff for verification of the physical part of the virtual experiments - DAQ system KSI 10 [6]. The latter is granted to the students for a remote training in the field of Electrical Measurement.

This Remote Virtual Laboratory is being constructed in the frame of THEIERE* Project as the main purpose was the efforts of the "Virtual Lab" group participants to be integrated in order to obtain a didactically structured distributed lab among the collaborative universities. The authors consider such an approach of teaching process organization as a possibility to make it more intensive and overcoming some constrains of the traditional teaching process concerning time and space limits in practical subjects as Electronics and Electrical Measurement in which practical experiments cannot be replaced by program simulators.



Base address: 0x300
Fast ADC range: ±10.00
Slow ADC range: ±4.00
DAC range: ±5.00
IRQ: 3

Fast ADC – Channel: 0, Gain: 1, Number of samples: 1
Triggering – Mode: Software, Freq: 1000 Hz
0 -> 0.59082031 (79)

Slow ADC – Channel: 0, Gain: 1 – 0.67203907
DAC – ○ 1 ● 2 – Voltage: 2.00000 – 1.99951172

Digital input – Port: 0, Value:
Digital output – Port: 0, Value: 0

U1,V 3 U2,V 1 U0,V 2.032453

Number	U1,V	U2,V	U0,V (exp)	U0,V (sim)
1	2.0	1.0	1.0742668	1.0
2	3.0	1.0	2.032453	2.0
3				
4				
5				
6				
7				
8				
9				
10				

Applet <UAmplifierApplet started

a) b)

Fig. 4. Virtual instruments as tools of a Remote Virtual Lab

Bibliography

1. Spoedler H.J.W Virtual Instrumentation and Virtual Environments, IEEE Instrumentation & Measurement Magazine, 1999, pp. 14-19.
2. Spoedler H.J.W, Ulings Virtual Instrumentation: A survey of Standards and their Interrelation, IEEE Technology Conference Ottawa, Canada, May 19-21, 1997, pp. 676-681.
3. Oppenheim A.V., A. Willsky, I. Young Signals and Systems, Prentice- Hall, Inc., 1983
4. Zheliazkova I., G. Georgiev, Organization of Measurement Systems in an Event Programming Environment, Automatica and Informatics Journal, N6,1999, pp.44-49.
5. Georgiev G., E.Stoyanov, S. Stefanova, D.Stoykov, H.Roth, Step by step towards the Virtual Lab organization, Int. Journal "Information Theories&Applications", 2002, vol.9, pp.153-159
6. DAQ system KSI 10, BSA, ITCR, Sofia, 1988
7. Pisani U., F.Cambiotti, N.Sala, F.Sanpriero, A Hypermedia solution for electronic instrumentation measurements practice, Proceedings of the 3rd International Conference on Computer Aided Engineering Education, 13-15 September, 1995, Bratislava, Slovaki, pp.144-149.
8. Majumdar S., C.Mazidar, D.Ray, On line CAI on Microprcessor Interfacing, Proceedings of the 3-rd International Conference on Computer Aided Engineering Eduvation, 1995, Bratislava, Slovakia,pp.144-149.
9. Ponta D., G.Donzellini and G.Parodi, "Practical electronics taught by hypertext: the WORKBENCH project", Int.J.Computers & Education 16(1), 1998, 127-132.
10. Ponta D., G.Donzellini, "Learning electronics with hypermedia and computer tools", Proceedings of the International Conference on Computer Aided Learning and Instruction in Science and Engineering, August 31, September 1-2,1997, Paris, France, pp.7-23.

Author information

G.S.Georgiev, G.T.Georgiev, S.L.Stefanova - Rousse University, 7017 Rousse, 8 Studentska Str. e-mail: gsg@ru.acad.bg

* Thematic Harmonisation in Electrical and Information EngineeRing in Europe,Project Nr. 10063-CP-1-2000-1-PT-ERASMUS-ETNE

AUTHOR INDEX OF THE IJ ITA VOL.10 / 2003

Volume 10 / 2003 of the IJ ITA is separated in 4 numbers:
Number 1: p.p.1-120; Number 2: p.p.121-240; Number 3: p.p.241-360; Number 4: p.p.361-480.

TABLE OF CONTENTS OF IJ ITA VOL.10

Number 1

Number 2

Number 3

TABLE OF CONTENTS OF IJ ITA VOL.10, NUMBER 4